



Sun Java™ System
Portal Server 6
管理ガイド

2005Q1

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-1198

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. は、この製品に含まれるテクノロジーに関する知的所有権を保持しています。特に限定されることなく、これらの知的所有権は <http://www.sun.com/patents> に記載されている 1 つ以上の米国特許および米国およびその他の国における 1 つ以上の追加特許または特許出願中のものが含まれている場合があります。

このソフトウェアは SUN MICROSYSTEMS, INC. の機密情報と企業秘密を含んでいます。SUN MICROSYSTEMS, INC. の書面による許諾を受けることなく、このソフトウェアを使用、開示、複製することは禁じられています。

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

この配布には、第三者が開発したソフトウェアが含まれている可能性があります。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd が独占的にライセンスしている米国およびその他の国における登録商標です。

Sun、Sun Microsystems、Sun のロゴマーク、Java、Solaris、JDK、Java Naming and Directory Interface、JavaMail、JavaHelp、J2SE、iPlanet、Duke のロゴマーク、Java Coffee Cup のロゴ、Solaris のロゴ、SunTone 認定ロゴマークおよび Sun ONE ロゴマークは、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

Legato および Legato のロゴマークは Legato Systems, Inc. の商標であり、Legato NetWorker は同社の商標または登録商標です。

Netscape Communications Corp のロゴマークは Netscape Communications Corporation の商標または登録商標です。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカルユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

この製品は、米国の輸出規制に関する法規の適用および管理下にあり、また、米国以外の国の輸出および輸入規制に関する法規の制限を受ける場合があります。核、ミサイル、生物化学兵器もしくは原子力船に関連した使用またはかかる使用者への提供は、直接的にも間接的にも、禁止されています。このソフトウェアを、米国の輸出禁止国へ輸出または再輸出すること、および米国輸出制限対象リスト (輸出が禁止されている個人リスト、特別に指定された国籍者リストを含む) に指定された、法人、または団体に輸出または再輸出することは一切禁止されています。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

目次

手順一覧	17
はじめに	23
対象読者	23
お読みになる前に	23
内容の紹介	24
表記上のルール	25
表記ルール	25
記号	26
デフォルトのパスおよびファイル名	27
シェルプロンプト	28
関連マニュアル	28
マニュアルセットに含まれるマニュアル	28
その他の Portal Server マニュアル	29
その他のサーバーマニュアル	30
Sun リソースへのオンラインによるアクセス	30
Sun テクニカルサポートの連絡先	30
関連サードパーティ Web サイトの参照	31
コメントの送付方法	31
第 1 章 Sun Java System Portal Server の管理の概要	33
アーキテクチャの概要	33
ポータルアクセスの概要	35
サービス設定の概要	37
Access Manager サービス	37
Portal Server サービス	38
デスクトップ	38

リライタ	38
検索エンジン	38
NetMail	39
WSRP	39
SSO アダプタ	39
登録	39
Portal Server サービスのための設定の仕組み	40
管理の概要	42
Access Manager コンソールの使用	42
コマンドラインユーティリティの使用	44

第 I 部 Sun Java System Portal Server の設定 45

第 2 章 インストール後の設定	47
Portal Server 設定プログラム	47
設定プログラムの実行	47
ローカライズ環境での設定プログラムの実行	48
設定チェックリスト	49
Portal Server と Secure Remote Access	49
ゲートウェイ	52
Netlet プロキシ	54
リライタプロキシ	55
Web コンテナのチェックリスト	57
Sun Java System Web Server のチェックリスト	58
Sun Java System Application Server のチェックリスト	58
BEA WebLogic Server のチェックリスト	59
IBM WebSphere Application Server のチェックリスト	61
Portal Server インストール後の作業	62
Portal Server	62
Sun Java System Web Server	62
Sun Java System Application Server	63
BEA WebLogic Server	64
IBM WebSphere Application Server	65
Secure Remote Access	65
ゲートウェイ	66
Netlet プロキシとリライタプロキシ	67
Portal Server インストールの確認	68
Portal Server 管理コンソールおよびデスクトップへのアクセス	68
Sun Java System Access Manager 管理コンソールにアクセスする	68
Portal Server デスクトップにアクセスする	68
ゲートウェイのインストールの確認	69

第 3 章 サーバーのインスタンスの作成および削除	71
サーバーのインスタンスを作成する	71
サーバーのインスタンスを削除する	72
第 4 章 セキュリティ保護された外部 LDAP Directory Server を使用する場合の Portal Server の設定	75
Directory Server の SSL 設定	76
信頼データベースを作成する	77
password.conf ファイルを使用する	78
root 認証局 (CA) 証明書をインストールする	79
Directory Server との通信に Access Manager の SSL 利用を有効にする	80
第 5 章 root 以外のユーザーとして Portal Server を実行するための設定	83
第 II 部 Sun Java System Portal Server の管理	87
第 6 章 認証、ユーザー、およびサービスの管理	89
Sun Java System Access Manager の概要	90
Access Manager の機能の概要	90
Portal Server 3.0 と Portal Server 6.2 の比較	91
Portal Server 6.0 と Portal Server 6.2 の比較	95
Access Manager の制約	96
Access Manager インタフェース	97
Access Manager 管理コンソール	97
Access Manager コマンド行	97
Access Manager 管理コンソールへのログイン	98
IP アドレスを使用した管理コンソールへのログインの設定	99
基本情報の表示	99
Portal Server の起動と停止	100
Access Manager サービスの管理	101
インストールおよび Sun Java System Web Server パッケージ	101
ユーザー管理	101
シングルサインオン / 認証	101
サービス管理	102
Portal Server ユーザーの管理	102
組織、サブ組織、およびロールの設計	103
組織およびサブ組織	103
ロール	104
ユーザー	104
シナリオ 1: サブ組織とロールから構成される階層構造	104
シナリオ 2: フラットなツリー構造	106

組織およびサブ組織の新規作成	108
新規の組織またはサブ組織を作成する	109
サービスを追加する	110
サービスのテンプレートを作成する	110
新規ユーザーを追加する	111
ユーザーにサービスを追加する	112
新規ロールを作成する	113
ロールをユーザーに割り当てる	113
既存のユーザーの Portal Server へのアクセスを有効化する	114
デフォルト組織のユーザーを有効化する	114
デフォルト以外の組織のユーザーを有効化する	119
ポータル組織の迅速な新規作成	121
認証の設定	124
認証レベルによる認証	125
認証メニューを設定する	126
認証順序を設定する	127
外部ディレクトリに LDAP 認証を設定する	128
匿名認証の設定	129
匿名認証を設定する (匿名ユーザーセッション方式)	130
匿名認証を設定する (認証なしアクセス)	131
連携ユーザー用の Portal Server の設定	133
連携ユーザーを設定する	133
連携ユーザーの認証なしアクセスを設定する	133
UNIX 認証を設定する	134
組織レベルの UNIX 認証を設定する	135
Portal Server によるポリシー管理の使用法の概要	136
ピアまたはサブ組織のポリシーサービスを追加する	137
ピアまたはサブ組織の参照ポリシーを作成する	138
ピアまたはサブ組織の標準のポリシーを作成する	139
Portal Server デスクトップへのログイン	140
サンプルのポータルデスクトップにログインする	140
サブ組織にログインする	140
匿名認証を使用してログインする	140
ロギングの管理	141
第 7 章 管理の委任の設定	143
管理の委任の概要	143
委任管理ロール	144
管理の委任モデルの定義	146
管理の委任の設定	147
Role Administrator Roles の ACI 設定の定義	147
コマンド行を使用して ACI を定義する	148
管理コンソールを使用して ACI を定義する	151

委任モデルのための管理ロールの新規作成	152
ロール管理者ロールを割り当てる	153
ロール管理者ロールで追加制限を設定する	154
第 8 章 ポータルデスクトップサービスの管理	157
デスクトップの概要	157
デスクトップ用語集	157
ポータルデスクトップのアーキテクチャとコンテナ階層	158
ユーザー定義のチャンネル	160
ポータルデスクトッププロバイダ	161
ポータルデスクトップサービス	162
サンプルデスクトップ	162
ポータルデスクトップのカスタマイズ	162
チャンネルのホット配備の概要	163
プロバイダアーカイブの概要	163
ポータルデスクトップサービスの管理	164
サブ組織のポリシーサービスを追加する	165
サブ組織の参照ポリシーを作成する	166
サブ組織に標準のポリシーを作成する	168
ログインユーザーをポータルデスクトップ URL に正しくリダイレクトする	169
ログインユーザーをポータルデスクトップ URL にリダイレクトする (グローバル)	170
ポータルデスクトップサービス属性の値を修正する	170
ポータルデスクトップサービス属性の値を修正する (グローバル)	171
サンプルポータルデスクトップにアクセスする	172
デスクトップログを確認する	172
ポートレットの管理	173
ポートレットからチャンネルを作成する	174
特定コンテナのポートレットからチャンネルを作成する	174
コンテナにポートレットチャンネルを追加する	175
ポートレットチャンネルの設定とプロパティを編集する	176
par ファイルの管理	178
新規の par ファイルを作成する	178
既存の par ファイルを修正する	179
par ファイルを配備する	180
第 9 章 Web Services for Remote Portlets (WSRP) サービスの管理	181
WSRP 標準の概要	181
WSRP プロデューサの管理	182
WSRP プロデューサインスタンスを追加する	183
WSRP プロデューサインスタンスを編集する	184
WSRP コンシューマ登録を追加する	186
WSRP コンシューマ登録を編集する	187

すべての WSRP プロデューサを無効にする	188
WSRP コンシューマの管理	189
リモートポートレットチャンネルを作成する	189
WSRP コンシューマの一般プロパティを編集する	190
設定済みの WSRP プロデューサを追加する	191
設定済みの WSRP プロデューサを編集する	192
すべての WSRP コンシューマを無効にする	193
標準ユーザープロファイルのマッピングを編集する	194
コンシューマ名を指定する	194
第 10 章 ディスプレイプロファイルの管理	195
ディスプレイプロファイルの概要	195
プロフィールと管理コンソールの表示	197
ディスプレイプロファイルドキュメントの構造	198
ディスプレイプロファイルルートオブジェクト	198
プロバイダオブジェクト	199
チャンネルオブジェクト	199
コンテナオブジェクト	200
ディスプレイプロファイルオブジェクトの集約	202
ディスプレイプロファイルオブジェクトの検索	203
ディスプレイプロファイルのプロパティ	204
ディスプレイプロファイルのプロパティのタイプ	204
ドキュメントタイプの定義要素の属性	205
ディスプレイプロファイルのプロパティの指定	207
入れ子のプロパティ	207
名前を付けられていないプロパティ	207
条件付きプロパティ	208
ディスプレイプロファイルのプロパティの伝達	210
ディスプレイプロファイルドキュメントの優先順位	213
ドキュメントの優先順位の例 1	214
ドキュメントの優先順位の例 2	215
ディスプレイプロファイルドキュメントの優先順位設定の要約	217
ディスプレイプロファイルのマージ動作	218
マージプロセスの仕組み	219
ディスプレイプロファイルのマージタイプ	219
remove の例 : remove マージによるコンテナの選択チャンネルリストの修正	220
replace の例 : replace マージによるすべてのユーザーの表示からのチャンネルの削除	222
fuse の例 : fuse マージによるロールベースのチャンネルリストの作成	223
マージロック	224
マージロックの例 : lock マージによるすべてのユーザーに対するプロパティ値の強制	225
マージロックの例 : lock マージによるすべてのユーザー表示からのチャンネルの強制削除	225
ディスプレイプロファイルと Sun Java System Access Manager	226
ディスプレイプロファイルの管理	227

デフォルトのディスプレイプロファイルドキュメント	229
ディスプレイプロファイルのロード	230
ディスプレイプロファイルをロードする (管理コンソール)	230
ディスプレイプロファイルをロードする (コマンド行)	231
ディスプレイプロファイルをダウンロードおよびアップロードする	232
ディスプレイプロファイル全体を表示する	233
ディスプレイプロファイルを削除する	233
「チャンネルおよびコンテナを管理」リンクを使用したチャンネルの管理	234
チャンネルとコンテナ管理のデフォルトプロバイダ	234
チャンネルの追加	234
シンプル Web サービスプロバイダ	235
事前に設定されている Web サービスチャンネル	236
設定可能な Web サービスチャンネル	236
新規コンテナチャンネル	237
チャンネルまたはコンテナチャンネルを作成する	238
チャンネルまたはコンテナチャンネルのプロパティを変更する	239
チャンネルまたはコンテナチャンネルを削除する	240
コンテナの管理	240
dpadmin コマンドの使用	241
dpadmin コマンドの使用に関するガイドライン	243
ディスプレイプロファイルの修正	244
ディスプレイプロファイルのエラーメッセージの概要	244
ディスプレイプロファイルオブジェクトを表示する	245
コンテナのチャンネルを置換する	245
チャンネルのプロパティを置換する	246
コンテナにチャンネルを追加する	247
コレクションにプロパティを追加する	248
コレクションプロパティを追加する	249
チャンネルまたはコンテナからプロパティを削除する	250
プロバイダを削除する	250
コンテナからチャンネルを削除する	251
ディスプレイプロファイルドキュメントの優先順位を変更する	251
チャンネルをコンテナで使用できるようにする	252
チャンネルをコンテナで使用不可にする	252
コンテナの使用可能なチャンネルリストからチャンネルを選択する	253
コンテナの使用可能なチャンネルリストからチャンネルを選択解除する	253
ディスプレイプロファイルテキストウィンドウの使用	254
ディスプレイプロファイルテキストウィンドウにアクセスする	254

第 11 章 NetMail サービスの管理	257
NetMail サービスの概要	257
NetMail サービスの管理	258
ピアまたはサブ組織のポリシーサービスを追加する	258

サブ組織の参照ポリシーを作成する	260
サブ組織に標準のポリシーを作成する	261
NetMail のサービス属性の変更 (特定の組織)	262
NetMail のサービス属性の変更 (すべての組織)	262
新しいウィンドウを開くように NetMail Lite を設定する	263
リモートアドレス帳の使用 (LDAP)	264
第 12 章 リライターサービスの管理	267
リライターサービスの概要	267
相対 URL の絶対 URL への拡張	268
URLScrapperProvider の制限	268
既存 URL へのゲートウェイ URL の追加	269
サポートされる URL	270
リライタルールとルールセットの定義	270
HTML コンテンツのルール	272
HTML コンテンツの属性ルール	272
HTML コンテンツの JavaScript トークンルール	273
HTML コンテンツのフォームルール	274
HTML コンテンツのアプレットルール	274
JavaScript コンテンツのルール	275
JavaScript 変数	275
JavaScript 関数パラメータ	277
XML コンテンツのルール	278
XML のタグテキスト	278
XML の属性	279
リライターサービスの管理	280
リライター URLScrapperProvider を SSL 用に設定する	280
デフォルトテンプレートからルールセットを新規作成する	281
既存のルールセットを編集する	282
ルールセットをダウンロードする	282
ルールセットをアップロードする	283
既存のルールセットを削除する	284
デフォルトのルールセットを復元する	284
第 13 章 検索エンジンサービスの管理	285
検索エンジンサービスの概要	285
検索データベース	286
検索ロボット	286
データベース分類カテゴリ	287
検索チャネルの設定	288
検索サーバーの初期設定	288
検索 URL を定義する	290

検索エンジンの管理	291
検索エンジン操作の表示、管理、監視	291
基本設定を表示または管理する	291
詳細設定を表示または管理する	292
検索エンジンの状況を監視する	292
ロボットの管理	293
サイトの定義	293
ロボットでインデックスを作成するサイトを定義する	294
ロボットの巡回の制御	294
ロボットの巡回を制御する	295
ロボットデータのフィルタリング	296
新規フィルタ定義を作成する	297
既存のフィルタ定義を変更する	297
フィルタを有効または無効にする	298
インデックス作成の属性の定義	298
インデックス作成の属性を定義する	299
ロボットユーティリティの使用	299
サイトプローブユーティリティを実行する	300
シミュレータを実行する	300
ロボットのスケジュール設定	301
ロボットのスケジュールを設定する	301
データベースの管理	302
データベースへのインポート	302
インポートエージェントを作成する	303
既存のインポートエージェントを編集する	304
リソース記述の編集	304
リソース記述を編集する	304
データベーススキーマの編集	305
データベーススキーマを編集する	306
スキーマのエイリアスの定義	307
スキーマのエイリアスを定義する	307
データベース分析の表示	308
データベース分析情報を表示する	308
データベースのインデックスの再作成	309
データベースのインデックスを再作成する	309
データベースの期限切れ	310
データベースを期限切れにする	310
データベースの破棄	310
サーバーから期限切れのリソース記述を破棄する	311
データベースのパーティションの作成	311
データベース分類の管理	312
カテゴリの設定	312
サブカテゴリを作成する	312

カテゴリを更新する	313
カテゴリを削除する	314
分類ルールの定義	315
分類ルールを定義する	315
第 14 章 検索エンジンロボットの管理	317
検索エンジンロボットの概要	317
ロボットの動作の仕組み	318
ロボット設定ファイル	319
ロボットプロセスパラメータの設定	320
フィルタリング処理	320
フィルタ処理の段階	321
フィルタの構文	322
フィルタディレクティブ	323
フィルタの記述または修正	324
ユーザーが変更できるパラメータ	324
robot.conf ファイルの例	331
第 15 章 定義済みのロボットアプリケーション関数	333
ソースとデスティネーション	334
セットアップ段階で使用可能なソース	334
メタデータフィルタリング段階で使用可能なソース	334
データ段階で使用可能なソース	335
列挙、生成、およびシャットダウン段階で使用可能なソース	336
有効化パラメータ	336
セットアップ関数	337
filterrules-setup	337
setup-regex-cache	337
setup-type-by-extension	338
フィルタリング関数	339
filter-by-exact	339
filter-by-max	340
filter-by-md5	340
filter-by-prefix	341
filter-by-regex	341
filterrules-process	342
サポート関数のフィルタリング	342
assign-source	343
assign-type-by-extension	343
clear-source	344
convert-to-html	345
copy-attribute	345

generate-by-exact	346
generate-by-prefix	347
generate-by-regex	347
generate-md5	348
generate-rd-expires	348
generate-rd-last-modified	349
rename-attribute	349
列挙関数	350
enumerate-urls	350
enumerate-urls-from-text	351
生成関数	351
extract-full-text	352
extract-html-meta	352
extract-html-text	353
extract-html-toc	353
extract-source	354
harvest-summarizer	355
シャットダウン関数	355
filterrules-shutdown	355
第 16 章 登録サービスの管理	357
概要	357
登録サービスの管理	358
ルートレベル	358
組織レベル	358
組織ユーザーレベル	358
ルートレベルで登録サービスを定義する	359
組織レベルで登録サービスを定義する	359
ユーザーの登録サービスを管理する	360
登録チャンネルの使用	362
カテゴリに登録する	362
ディスカッションに登録する	363
検索を保存する	363
ディスカッション	364
ディスカッションの概要	364
DiscussionProvider	364
DiscussionProvider のディスプレイプロファイル XML コード	366
DiscussionProvider の管理	367
DiscussionLite チャンネル	367
Discussions チャンネル	368
チャンネルの管理と使用	370
DiscussionProvider チャンネルの管理	370
DiscussionProvider からチャンネルを作成する	371

DiscussionProvider サンプルチャンネルの使用	372
新しいディスカッションを開始する	372
第 17 章 通信チャンネルの設定	373
通信チャンネルの概要	374
通信チャンネルでサポートされるソフトウェア	374
インストーラと通信チャンネル	375
Sun Java System Portal Server インストーラが実行するタスク	375
複数インスタンスでの配備	376
通信チャンネルの設定タスク	377
メールアプリケーションとカレンダーアプリケーションへのアクセスを有効化する	377
Messaging Server 用の ipsecurity を無効にする	378
Calendar Server 用の ipsecurity を無効にする	378
デフォルトの組織のサービスを設定する	379
通信チャンネルの設定に関する情報	379
Instant Messaging チャンネルの設定	380
アドレス帳チャンネルの設定	387
エンドユーザーとしてチャンネルを設定する	388
アプリケーション設定の編集: 通信チャンネル編集ページの設定	391
編集ページのディスプレイプロファイルの属性	391
編集ページの HTML テンプレート	393
ディスプレイプロファイルの例	394
通信チャンネルのタイプとしてエンドユーザーが複数インスタンスを設定できるようにする	395
管理者のプロキシ認証: エンドユーザーによる信用情報の設定の禁止	396
プロキシ認証の設定方法の概要	397
プロキシ認証と SSO (シングルサインオン) アダプタテンプレート	397
プロキシ認証と通信サーバー	399
認証不要ポータルデスクトップ用に読み取り専用通信チャンネルを設定する	400
読み取り専用通信チャンネルに関する情報と注意点	400
カレンダーユーザーを設定する	400
読み取り専用通信チャンネルを設定する	401
Microsoft Exchange Server または IBM Lotus Notes の設定	404
アドレス帳、カレンダー、メール用に Microsoft Exchange 5.5 Server を設定する	404
アドレス帳、カレンダー、メール用に Microsoft Exchange 2000 Server を設定する	406
ocxhost.exe をアンインストールする	411
アドレス帳、カレンダー、メール用に Lotus Domino Server を設定する	411
Lotus Notes の設定	413
デフォルト組織の下に新規ユーザーを作成する	418
HTTPS が有効な Messaging Server とメールプロバイダが連携するように設定する	419
Web コンテナに関する情報と注意点	419
HTTPS が有効な Messaging Server とメールプロバイダが連携するように設定する	419

第 18 章 Portal Server システムの管理	425
Secure Sockets Layer (SSL) の設定	425
Portal Server で SSL を設定する	426
SSL を使用する既存の Portal Server インストールを変更する	427
SSL を使用するよう Portal Server インスタンスを設定する	428
Portal Server の設定のバックアップと復元	430
Portal Server の設定をバックアップする	430
Portal Server の設定を復元する	431
Portal Server のネットワーク設定の変更	432
複数の UI ノードインストールの管理	432
サーバーリストに Portal Server を追加する	433
HTTP プロキシを使用する Portal Server のインスタンスの設定	433
Portal Server ログの管理	434
ファイルにログを記録するように設定する	434
データベースにログを記録するように設定する	435
Portal Server のデバッグ	435
Sun Java System Access Manager のデバッグレベルを設定する	435
第 III 部 Sun Java System Portal Server のチューニング	437
第 19 章 Portal Server のチューニング	439
紹介	440
チューニング方法	441
Solaris のチューニング	441
カーネルのチューニング	441
TCP パラメータのチューニング	442
Sun Java System Access Manager のチューニング	443
Directory Server の接続プール	443
LDAP 認証サービス	443
Sun Java System Access Manager サービス設定パラメータ	443
Sun Java System Directory Server のチューニング	444
Sun Java System Web Server 6.1 のチューニング	446
Sun Java System Application Server 7.0 のチューニング	447
その他の Sun Java System Application Server パラメータの設定によるゲートウェイの 信頼性向上	449
Portal Server デスクトップのチューニング	451
付録 A SSO アダプタテンプレートと設定	453
シングルサインオンアダプタの概要	453
SSO アダプタ属性ページ	454
SSO アダプタテンプレートを作成する	455

SSO アダプタの設定を作成する	455
SSO アダプタテンプレートのプロパティを編集する	456
SSO アダプタの設定プロパティを編集する	456
用語集	457
索引	459

手順一覧

Sun Java System Access Manager 管理コンソールにアクセスする	68
Portal Server デスクトップにアクセスする	68
サーバーのインスタンスを作成する	71
サーバーのインスタンスを削除する	72
Directory Server の SSL 設定	76
信頼データベースを作成する	77
password.conf ファイルを使用する	78
root 認証局 (CA) 証明書をインストールする	79
Directory Server との通信に Access Manager の SSL 利用を有効にする	80
シナリオ 2: フラットなツリー構造	106
組織およびサブ組織の新規作成	108
新規の組織またはサブ組織を作成する	109
サービスを追加する	110
サービスのテンプレートを作成する	110
新規ユーザーを追加する	111
ユーザーにサービスを追加する	112
新規ルールを作成する	113
ルールをユーザーに割り当てる	113
既存のユーザーの Portal Server へのアクセスを有効化する	114
デフォルト組織のユーザーを有効化する	114
デフォルト以外の組織のユーザーを有効化する	119
ポータル組織の迅速な新規作成	121
認証メニューを設定する	126
認証順序を設定する	127
外部ディレクトリに LDAP 認証を設定する	128
匿名認証の設定	129
匿名認証を設定する (匿名ユーザーセッション方式)	130

匿名認証を設定する (認証なしアクセス)	131
UNIX 認証を設定する	134
ピアまたはサブ組織のポリシーサービスを追加する	137
ピアまたはサブ組織の参照ポリシーを作成する	138
ピアまたはサブ組織の標準のポリシーを作成する	139
サンプルのポータルデスクトップにログインする	140
サブ組織にログインする	140
匿名認証を使用してログインする	140
Role Administrator Roles の ACI 設定の定義	147
コマンド行を使用して ACI を定義する	148
管理コンソールを使用して ACI を定義する	151
委任モデルのための管理ロールの新規作成	152
ロール管理者ロールを割り当てる	153
ロール管理者ロールで追加制限を設定する	154
サブ組織のポリシーサービスを追加する	165
サブ組織の参照ポリシーを作成する	166
サブ組織に標準のポリシーを作成する	168
ログインユーザーをポータルデスクトップ URL に正しくリダイレクトする	169
ログインユーザーをポータルデスクトップ URL にリダイレクトする (グローバル)	170
ポータルデスクトップサービス属性の値を修正する	170
ポータルデスクトップサービス属性の値を修正する (グローバル)	171
サンプルポータルデスクトップにアクセスする	172
デスクトップログを確認する	172
ポートレットからチャンネルを作成する	174
特定コンテナのポートレットからチャンネルを作成する	174
コンテナにポートレットチャンネルを追加する	175
新規の par ファイルを作成する	178
既存の par ファイルを修正する	179
par ファイルを配備する	180
WSRP プロデューサインスタンスを追加する	183
WSRP プロデューサインスタンスを編集する	184
WSRP コンシューマ登録を追加する	186
WSRP コンシューマ登録を編集する	187
すべての WSRP プロデューサを無効にする	188
リモートポートレットチャンネルを作成する	189
WSRP コンシューマの一般プロパティを編集する	190
設定済みの WSRP プロデューサを追加する	191

設定済みの WSRP プロデューサを編集する	192
すべての WSRP コンシューマを無効にする	193
標準ユーザープロファイルのマッピングを編集する	194
コンシューマ名を指定する	194
ディスプレイプロファイルをロードする (管理コンソール)	230
ディスプレイプロファイルをロードする (コマンド行)	231
ディスプレイプロファイルをダウンロードおよびアップロードする	232
チャンネルまたはコンテナチャンネルを作成する	238
チャンネルまたはコンテナチャンネルのプロパティを変更する	239
チャンネルまたはコンテナチャンネルを削除する	240
ディスプレイプロファイルのエラーメッセージの概要	244
ディスプレイプロファイルオブジェクトを表示する	245
コンテナのチャンネルを置換する	245
チャンネルのプロパティを置換する	246
コンテナにチャンネルを追加する	247
コレクションにプロパティを追加する	248
コレクションプロパティを追加する	249
チャンネルまたはコンテナからプロパティを削除する	250
プロバイダを削除する	250
コンテナからチャンネルを削除する	251
ディスプレイプロファイルドキュメントの優先順位を変更する	251
チャンネルをコンテナで使用できるようにする	252
チャンネルをコンテナで使用不可にする	252
コンテナの使用可能なチャンネルリストからチャンネルを選択する	253
コンテナの使用可能なチャンネルリストからチャンネルを選択解除する	253
ディスプレイプロファイルテキストウィンドウにアクセスする	254
ピアまたはサブ組織のポリシーサービスを追加する	258
サブ組織の参照ポリシーを作成する	260
サブ組織に標準のポリシーを作成する	261
NetMail のサービス属性の変更 (特定の組織)	262
NetMail のサービス属性の変更 (すべての組織)	262
新しいウィンドウを開くように NetMail Lite を設定する	263
リモートアドレス帳の使用 (LDAP)	264
リライタ URLScrapperProvider を SSL 用に設定する	280
デフォルトテンプレートからルールセットを新規作成する	281
既存のルールセットを編集する	282
ルールセットをダウンロードする	282

既存のルールセットを削除する	284
デフォルトのルールセットを復元する	284
検索サーバーの初期設定	288
検索 URL を定義する	290
検索エンジン操作の表示、管理、監視	291
基本設定を表示または管理する	291
詳細設定を表示または管理する	292
検索エンジンの状況を監視する	292
サイトの定義	293
ロボットでインデックスを作成するサイトを定義する	294
ロボットの巡回の制御	294
ロボットの巡回を制御する	295
ロボットデータのフィルタリング	296
新規フィルタ定義を作成する	297
既存のフィルタ定義を変更する	297
フィルタを有効または無効にする	298
インデックス作成の属性の定義	298
インデックス作成の属性を定義する	299
ロボットユーティリティの使用	299
サイトブローユーティリティを実行する	300
シミュレータを実行する	300
ロボットのスケジュール設定	301
ロボットのスケジュールを設定する	301
データベースへのインポート	302
インポートエージェントを作成する	303
既存のインポートエージェントを編集する	304
リソース記述の編集	304
リソース記述を編集する	304
データベーススキーマの編集	305
データベーススキーマを編集する	306
スキーマのエイリアスを定義する	307
データベース分析情報を表示する	308
データベースのインデックスの再作成	309
データベースのインデックスを再作成する	309
データベースの期限切れ	310
データベースを期限切れにする	310
データベースの破棄	310

サーバーから期限切れのリソース記述を破棄する	311
データベースのパーティションの作成	311
カテゴリの設定	312
サブカテゴリを作成する	312
カテゴリを更新する	313
カテゴリを削除する	314
分類ルールの定義	315
分類ルールを定義する	315
ルートレベルで登録サービスを定義する	359
組織レベルで登録サービスを定義する	359
ユーザーの登録サービスを管理する	360
カテゴリに登録する	362
ディスカッションに登録する	363
検索を保存する	363
DiscussionProvider からチャンネルを作成する	371
新しいディスカッションを開始する	372
Messaging Server 用の ipsecurity を無効にする	378
Calendar Server 用の ipsecurity を無効にする	378
カレンダーユーザーを設定する	400
読み取り専用通信チャンネルを設定する	401
アドレス帳、カレンダー、メール用に Microsoft Exchange 5.5 Server を設定する	404
アドレス帳、カレンダー、メール用に Microsoft Exchange 2000 Server を設定する	406
ocxhost.exe をアンインストールする	411
アドレス帳、カレンダー、メール用に Lotus Domino Server を設定する	411
HTTPS が有効な Messaging Server とメールプロバイダが連携するように設定する	419
Portal Server で SSL を設定する	426
SSL を使用する既存の Portal Server インストールを変更する	427
SSL を使用するように Portal Server インスタンスを設定する	428
Portal Server の設定をバックアップする	430
Portal Server の設定を復元する	431
サーバーリストに Portal Server を追加する	433
ファイルにログを記録するように設定する	434
データベースにログを記録するように設定する	435
Sun Java System Access Manager のデバッグレベルを設定する	435
SSO アダプタテンプレートを作成する	455
SSO アダプタの設定を作成する	455
SSO アダプタテンプレートのプロパティを編集する	456

SSO アダプタの設定プロパティを編集する	456
-----------------------------	-----

はじめに

対象読者

このマニュアルは、Portal Server のインストール、管理、および設定を担当するユーザーを対象にしています。

お読みになる前に

Portal Server の管理を行う前に、次の概念について精通していることが必要です。

- Solaris™ の基本管理手順
- LDAP
- Sun Java™ System Directory Server (旧 Sun ONE Directory Server)
- Sun Java™ System Access Manager (旧 Sun ONE Identity Server および Sun Java System Identity Server)

使用している Web コンテナに応じて、次の概念の 1 つ以上について精通する必要があります。

- Sun Java™ System Web Server (旧 Sun ONE Web Server)
- Sun Java System Application Server
- BEA WebLogic Server™ 8.1 SP2
- IBM WebSphere® 5.1

注 Sun™ Java System 製品ファミリーは、以前の Sun™ ONE ブランドから名称変更されたものです。

内容の紹介

このマニュアルには、次の章と付録があります。

表 1 章と付録のリスト

章	説明
はじめに	(本章)
第 1 章「Sun Java System Portal Server の管理の概要」	Portal Server のアーキテクチャ、プロトコル、およびインタフェースについて説明します。また、製品の管理およびカスタマイズの概要について説明します。
第 2 章「インストール後の設定」	インストール後の設定作業を実行する手順について説明します。
第 3 章「サーバーのインスタンスの作成および削除」	Portal Server のインスタンスを作成および削除するための手順について説明します。
第 4 章「セキュリティ保護された外部 LDAP Directory Server を使用する場合の Portal Server の設定」	セキュアな外部 LDAP サーバーを使用する Portal Server をセットアップするための手順について説明します。
第 5 章「root 以外のユーザーとして Portal Server を実行するための設定」	root 以外のユーザーとして実行する際に Portal Server を設定する手順について説明します。
第 6 章「認証、ユーザー、およびサービスの管理」	Sun Java System Identity Server を使用して、認証、ユーザー、およびサービスを管理する方法を説明します。
第 7 章「管理の委任の設定」	Portal Server の委任管理を設定する方法を説明します。
第 8 章「ポータルデスクトップサービスの管理」	Portal Server デスクトップサービスを管理する方法について説明します。
第 9 章「Web Services for Remote Portlets (WSRP) サービスの管理」	Web Services for Remote Portlets (WSRP) の使用とその手順について説明します。
第 10 章「ディスプレイプロファイルの管理」	Portal Server ディスプレイプロファイルコンポーネントを管理する方法について説明します。
第 11 章「NetMail サービスの管理」	NetMail サービスを管理する方法について説明します。
第 12 章「リライトサービスの管理」	リライトサービスを管理する方法について説明します。
第 13 章「検索エンジンサービスの管理」	検索エンジンサービスの設定および管理方法について説明します。

表 1 章と付録のリスト (続き)

章	説明
第 14 章「検索エンジンロボット の管理」	検索エンジンロボットと、それに対応する設定ファイルについて説明します。
第 15 章「定義済みのロボットア プリケーション関数」	事前定義のロボットアプリケーション関数について説明します。これらの関数を使用して、フィルタ定義を作成および変更できます。
第 16 章「登録サービスの管理」	登録サービスの設定および管理方法について説明します。
第 17 章「通信チャネルの設定」	Portal Server の通信チャネルについて説明します。
第 18 章「Portal Server システムの 管理」	Portal Server システムの管理に関連するさまざまな管理作業について説明します。
第 19 章「Portal Server のチューニ ング」	Portal Server に含まれている perftune スクリプトについて説明します。
付録 A「SSO アダプタテンプレー トと設定」	Portal Server の通信チャネルのリファレンスです。

表記上のルール

ここでは、このマニュアルで使用されている表記上のルールを説明します。

表記ルール

次の表で、このマニュアルで使用されているさまざまな文字表記について説明します。

表 2 表記ルール

書体	意味	例
AaBbCc123 (モノスペース)	API および言語要素、HTML タグ、Web サイト URL、コマンド名、ファイル名、ディレクトリパス名、画面上のコンピュータ出力、サンプルコード。	.login ファイルを編集します。 ls -a を使用して、すべてのファイルを表示します。 % You have mail.
AaBbCc123 (太字のモノスペース)	ユーザーが入力する文字を、画面上のコンピュータ出力とは区別する場合。	% su パスワード :

表 2 表記ルール (続き)

書体	意味	例
<i>AaBbCc123</i> (イタリック)	コマンドまたはパス名で、実際の名前または値によって置き換えられる部分。	これらは <i>class</i> オプションと呼ばれています。 ファイルは、 <i>install-dir/bin</i> ディレクトリに配置されています。

記号

次の表は、このマニュアルで使用されている記号の表記ルールについて説明します。

表 3 記号の表記ルール

記号	説明	例	意味
[]	任意のコマンドオプションが含まれています。	ls [-l]	-l オプションは必須ではありません。
{ }	必須コマンドオプションの選択項目が含まれています。	-d {y n}	-d オプションは、y 引数または n 引数のいずれかを使用する必要があります。
-	同時に複数のキーストロークを行います。	Control-A	Control キーを押しながら、A キーを押します。
+	連続で複数のキーストロークを行います。	Ctrl+A+N	Control キーを押して離してから、次のキーを押します。
>	グラフィカルユーザーインターフェイスでのメニュー項目の選択を示します。	ファイル > 新規 > テンプレート	「ファイル」メニューから、「新規」を選択します。「新規」サブメニューから、「テンプレート」を選択します。

デフォルトのパスおよびファイル名

次の表は、このマニュアルで使用されているデフォルトのパス名とファイル名を説明します。

表 4 デフォルトのパスおよびファイル名

用語	説明
<i>PortalServer-base</i>	<p>Portal Server 用のベースインストールディレクトリを表します。Portal Server 2005Q1 のデフォルトのベースインストールディレクトリと製品ディレクトリは、使用するプラットフォームに依存します。</p> <p>Solaris™ システム : /opt</p> <p>Linux システム : /opt/sun</p>
<i>AccessManager-base</i>	<p>Access Manager 用のベースインストールディレクトリを表します。Access Manager 2005Q2 のデフォルトのベースインストールディレクトリと製品ディレクトリは、使用するプラットフォームに依存します。</p> <p>Solaris™ システム : /opt/SUNWam</p> <p>Linux システム : /opt/sun/identity</p>
<i>DirectoryServer-base</i>	<p>Sun Java System Directory Server 用のベースインストールディレクトリを表します。特定のパス名については、製品マニュアルを参照してください。</p>
<i>ApplicationServer-base</i>	<p>Sun Java System Application Server 用のベースインストールディレクトリを表します。特定のパス名については、製品マニュアルを参照してください。</p>
<i>WebServer-base</i>	<p>Sun Java System Web Server、BEA WEblogic 8.1 SP2、または IBM WebSphere 用のベースインストールディレクトリを表します。特定のパス名については、製品マニュアルを参照してください。</p>

シェルプロンプト

次の表は、このマニュアルで使用されているシェルプロンプトについて説明します。

表 5 シェルプロンプト

シェル	プロンプト
UNIX または Linux 上の C シェル	<i>machine-name%</i>
UNIX または Linux 上の C シェルスーパーユーザー	<i>machine-name#</i>
UNIX または Linux 上の Bourne または Korn シェル	\$
UNIX または Linux 上の Bourne または Korn シェルスーパーユーザー	#
Microsoft Windows コマンド行	C: ¥

関連マニュアル

<http://docs.sun.com/app/docs?l=ja>SM Web サイトから、Sun の技術マニュアルにオンラインでアクセスできます。アーカイブをブラウズしたり、特定の書名またはサブジェクトで検索できます。

マニュアルセットに含まれるマニュアル

次の表は、Portal Server の主要マニュアルセット内に含まれるマニュアルの概要を説明します。

書名	説明
『Portal Server 配備計画ガイド』 http://docs.sun.com/db/doc/819-1206?l=ja	Sun Java System Portal Server ソフトウェアの計画および配備について説明します。
『Portal Server 管理ガイド』 http://docs.sun.com/db/doc/819-1198?l=ja	Access Manager 管理コンソールとコマンド行を使って Sun Java System Portal Server 6 を管理する方法を説明します。
『Portal Server Secure Remote Access 管理ガイド』 http://docs.sun.com/db/doc/819-1202?l=ja	Sun Java System Portal Server 6 Secure Remote Access を管理する方法を説明します。

書名	説明
『Portal Server リリースノート』 http://docs.sun.com/db/doc/819-1494?l=ja	製品がリリースされた後に入手可能です。今回のリリースの新しい点、既知の問題や制限、インストールの注意点、ソフトウェアまたはマニュアルに関する問題を報告する方法などの説明を含む、最新情報があります。
『Portal Server Technical Reference Guide』 http://docs.sun.com/db/doc/817-7696	Sun Java System Portal Server の技術的概念 (ディスプレイプロファイル、リライタ)、コマンド行ユーティリティ、タグライブラリ (ソフトウェア内)、ファイル (テンプレートおよび JSP など) に関する詳細情報を提供します。このガイドは、上記の重要な基礎知識についての唯一の情報源です。

その他の Portal Server マニュアル

その他の Portal Server マニュアルには、次のものが含まれます。

- Portal Server Desktop Customization Guide
<http://docs.sun.com/doc/817-5318>
- Portal Server Developer's Guide
<http://docs.sun.com/doc/817-5319>
- Portal Server Mobile Access Developer's Guide
<http://docs.sun.com/doc/817-6258>
- Portal Server Mobile Access Developer's Reference
<http://docs.sun.com/doc/817-6259>
- Portal Server Mobile Access 配備計画ガイド
<http://docs.sun.com/doc/819-1961?l=ja>
- Portal Server Mobile Access Tag Library Reference
<http://docs.sun.com/doc/817-6260>

その他のサーバーマニュアル

その他のサーバーマニュアルについては、次を参照してください。

- Directory Server マニュアル
http://docs.sun.com/app/docs/coll/DirectoryServer_04q2_ja
- Web Server マニュアル
http://docs.sun.com/app/docs/coll/S1_websvr61_ja
- Application Server マニュアル
http://docs.sun.com/app/docs/coll/s1_asseu3_ja
- Web Proxy Server マニュアル
<http://docs.sun.com/app/docs/prod/s1.webproxys#hic>

Sun リソースへのオンラインによるアクセス

製品ダウンロード、専門的サービス、パッチとサポート、追加の開発者情報については、次にアクセスしてください。

- ダウンロードセンター
<http://www.sun.com/software/download/>
- 専門的サービス
<http://www.sun.com/service/sunps/sunone/index.html>
- Sun エンタープライズサービス、Solaris パッチ、およびサポート
<http://sunsolve.sun.com/>
- 開発者情報
<http://developers.sun.com/prodtech/index.html>

Sun テクニカルサポートの連絡先

本製品に関する技術的質問で、製品マニュアル内に解決策がない場合は、次にアクセスしてください。<http://www.sun.com/service/contacting>

関連サードパーティ Web サイトの参照

Sun は、このマニュアルに記載されているサードパーティの Web サイトの有効性については責任を持ちません。Sun はそのようなサイトやリソースを通じて利用可能なコンテンツ、広告、製品、その他資料について、保証、責任、義務を負いません。Sun はそのようなサイトやリソースと通じて利用可能なコンテンツ、製品、サービスを、使用または信用したことによって、またはそれに関連して発生または申し立てられた、実際または疑惑のある損傷や損害に対しては責任または義務を負いません。

コメントの送付方法

Sun は、マニュアルの改善に努めており、ユーザーからのコメントおよび提案を歓迎いたします。

コメントを送付するには、<http://docs.sun.com> にアクセスして、「コメントの送付」をクリックしてください。オンラインフォームに、マニュアル名と Part No. を入力してください。パート番号は、マニュアルのタイトルページ、またはマニュアルの一番上に記載されている 7 桁または 9 桁の数字です。たとえば、このマニュアルのタイトルは『Sun Java System Portal Server 2005Q1 管理ガイド』で、パート番号は 819-1198 です。

コメントの送付方法

Sun Java System Portal Server の管理の概要

Sun Java™ System Portal Server 6 2005Q1 製品は、企業がさまざまなソースからコンテンツを引き出し、特定のユーザーまたはユーザーグループ用にコンテンツをパーソナライズし、これらの複数のソースのコンテンツを Web ブラウザなど特定ユーザーのデバイスに合わせて単一の出力形式に集約することができる一連の統合ソフトウェア製品です。

この章では、製品セットのアーキテクチャ、ポータルのエンドユーザーインターフェース、Portal Server ソフトウェアによって実装されるサービスの概要と設定方法、製品の管理に使用するツールについて簡単に説明します。この章で説明する内容は次のとおりです。

- [アーキテクチャの概要](#)
- [ポータルアクセスの概要](#)
- [サービス設定の概要](#)
- [管理の概要](#)

アーキテクチャの概要

Portal Server は Sun Java™ System アーキテクチャの一部です。Sun Java System アーキテクチャ内で Portal Server はコンテンツの配置、接続、集約、表示、通信、パーソナライズ、通知、および配信を行うテクノロジーを提供します。Sun Java System 内のコンテンツは、Web サービスにより提供されます。Portal Server 自体は Web サービスを提供しません。Portal Server はユーザーインターフェースを Web サービスに関連付けるための、また Web サービスをユーザーにとって便利なものにするメカニズムです。

Portal Server 製品のアーキテクチャは、統合可能なさまざまなソフトウェア製品で構成されます。これにより、Portal Server は内部コンポーネントだけでなく、サポートされている外部製品が提供する機能やサービスも活用できます。Portal Server 自体には、デスクトップ、NetMail、リライタ、検索などの内部コンポーネントがあります。サポートされている外部製品には、Sun Java™ System Web Server、Sun™ Java System Directory Server、および Sun Java™ System Access Manager (旧 Sun™ ONE Identity Server) があります。Portal Server は、Web アプリケーションコンテナ、ユーザー、サービス、ポリシー管理、認証とシングルサインオン、管理コンソール、ディレクトリスキーマとデータストレージ、プロトコルのサポートを Portal Server 製品自体に実装するのではなく、これらの外部製品で実装します。たとえば、Portal Server 製品はデフォルトの Web コンテナとして Sun Java System Web Server を使用します。

注 Portal Server は、次の Web コンテナをサポートします。Sun Java System Web Server、Sun Java™ System Application Server、IBM Websphere Application Server、さらに、BEA Weblogic Application Server も使用可能です。

Sun Java System Portal Server 2005Q1 は、次のコンポーネント製品を使用します。

- Java™ 2 SDK (J2SDK™), Standard Edition
- Web コンテナ
- Web コンテナ - Sun Java System Portal Server は、次の Web コンテナ上に配備できます。
 - Sun Java System Application Server 8.1
 - Sun Java System Web Server 6.1 SP4
 - BEA WebLogic Server™ 5.2 Patch 3
 - IBM WebSphere® Application Server 8.1 SP2
- Sun Java System Directory Server 5.1
- Sun Java System Access Manager 6.3

また、その他の Portal Server アドオンソフトウェアもインストールできます (Sun Java™ System Portal Server、Secure Remote Access など)。Portal Server のアーキテクチャについては、『Sun Java System Portal Server 6 2005Q1 配備ガイド』を参照してください。

ポータルアクセスの概要

通常、ユーザーは Web ブラウザを通じて、ポータルのホームページの URL を要求し、Sun Java System Access Manager 認証サービスを通じて認証することで、ポータルコンテンツにアクセスします。いったん認証されると、ユーザーに Portal Server デスクトップが表示されます。

35 ページの図 1-1 は、Portal Server のデスクトップの例を示しています。

図 1-1 Portal Server デスクトップの例

The screenshot displays the Sun Java System Portal Server desktop. At the top, there is a navigation bar with links for Home, Theme, Logout, Tabs, and Help. Below this, there are tabs for Front Page, Samples, and Search. The main content area is divided into several panels:

- ユーザー情報 (User Information):** Displays user details such as 'ようこそ ユーザー1', '最終更新日: 2006/03/07 15:47', '残りの時間 113 (分)', and '最大アイドル時間 30 (分)'.
- ブックマーク #2 (Bookmarks #2):** Includes a search bar and a list of bookmarked URLs: 'Sun ホームページ', 'Sun Java Enterprise System の詳細情報...', and 'Sun Software のホームページ'.
- Sun の情報 (Sun Information):** Provides links to 'Sun JavaTM Systems を参照...', 'Sun Software の最新情報...', and 'Sun Microsystems の最新情報...'.
- サンプル JSP チャネル (Sample JSP Channel):** Contains a section titled 'JSP プロバイダについて' and a configuration table:

JSP:	samplecontent.jsp
JSP の実際のパス:	/etc/opt/SUNWjps/desktop/sam
リクエストパラメータ:	action=content provider=WirelessDesktopDispatcher
セッションの属性:	なし
選択したユーザー属性:	名 (givenname) = 1 姓 (sn) ユーザー
- XML テストチャネル (XML Test Channel):** Displays a table of stock data for 'company22.com':

company22.com		NASDAQ, 15:47
終値	16.240000	始値 16.8
前日比	-0.85	前日終値 17.090000
前日比(%)	-4.97%	買い気配値 16.24
出来高	26786000	売り気配値 16.25
本日の最高値	16.99	52 週最高値 64.6562
本日の最安値	16.05	52 週最安値 12.85

デスクトップは、ポータルコンテンツとのプライマリユーザーインターフェースです。デスクトップサービスは、サブレット、プロバイダ API、さまざまなチャンネル、その他の各種サポート API、およびユーティリティを通じて実装されます。デスクトップは、プロバイダと呼ばれるプログラム上のエンティティを使用してコンテンツを生成します。コンテンツの 1 単位をチャンネルと呼びます。複数のチャンネルのコンテンツをコンテンツチャンネルに集約し、デスクトップ上でテーブルまたはタブなどのさまざまな形式に編集できます。ユーザーがポータルにアクセスすると、デスクトップはユーザーのコンテンツ生成に使用されたコンテンツプロバイダとチャンネルデータを格納したディスプレイプロファイルを参照します。ディスプレイプロファイルは、デスクトップでユーザーに表示される全体的なレイアウト、表示、あるいは組織を実際には定義しないので、誤解のないように注意してください。基本的に、ディスプレイプロファイルはチャンネルのプロパティ値を提供するだけです。実際には、デスクトップは Sun Java System Directory Server のさまざまなレベルまたはノード (最上位、組織、ロール、およびユーザーレベル) に LDAP 属性として格納された複数のディスプレイプロファイルを使用して、ユーザーに表示するコンテンツを決定します。各レベルのディスプレイプロファイルのプロパティ定義、およびプロパティ値の LDAP ノードへのアップロードには、XML ドキュメントが使用されます。実行時に、各レベルで定義されたディスプレイプロファイルのプロパティをマージして、ユーザーのディスプレイプロファイルが作成されます。ディスプレイプロファイルドキュメントは各レベルで定義できますが、各レベルにディスプレイプロファイルドキュメントを指定する必要はありません。

指定したクライアントタイプ (HTML または MAPI) に基づいて、特定のプロパティ値を保存、取得できるように、Sun Java System Portal Server ソフトウェアには次のプロパティ情報が含まれます。

- フィルタ条件を定義する条件付きプロパティ (208 ページの「[条件付きプロパティ](#)」を参照)。
- 認証が行われない場合のクライアントの管理方法を決定するための authlessState プロパティ (129 ページの「[匿名認証の設定](#)」を参照)。

サービス設定の概要

Portal Server は Sun Java™ Enterprise System アプリケーションであるため、そのサービスは Access Manager サービス管理システム (SMS) を使用して、定義および管理されます。サーバーに固有ではないサービス関連のデータは、SMS DTD (サービス管理システム文書型定義) に準拠した XML (Extensible Markup Language) ファイルを使用して定義されます。サーバー固有のデータは、個々のサーバーのローカルプロパティファイルに保存できます。各 Portal Server サービス (デスクトップ、Netmail、リライタ、検索) には、サービスに固有のデータを指定、変更するための独自の XML ファイルとプロパティファイルがあります。

Access Manager サービス

「[アーキテクチャの概要](#)」で説明したように、Portal Server は Portal Server 自体には含まれていない Sun Java System アーキテクチャのサポート製品を使用して、多くの機能およびサービスを実装します。特に、以前のバージョンの Portal Server が多くの管理機能を内部的に実装していた場合、Access Manager との統合により、Portal Server は Access Manager 製品の次の管理ツールおよびサービスを、効果的に活用することができます。

- 管理コンソール
- サービス管理
- ユーザー管理
- 認証 / シングルサインオン

Access Manager サービスの管理についての詳細は、[第 6 章「認証、ユーザー、およびサービスの管理」](#)を参照してください。

Portal Server サービス

標準の Access Manager サービス以外に、Portal Server は Access Manager 管理コンソールを使用して、内部サービス (デスクトップ、NetMail、リライタ、検索) を管理します。

デスクトップ

前の節で説明したように、デスクトップは Portal Server のプライマリエンドユーザーインタフェースを提供します。デスクトップは、プロバイダアプリケーションプログラミングインタフェース (PAPI) を使用して拡張可能な、コンテンツ集約のためのメカニズムを提供します。デスクトップには、コンテナ階層と、特定のチャンネルを構築するための基本構築ブロックとを有効にするさまざまなプロバイダが表示されます。コンテンツプロバイダとチャンネルデータを保存する場合、デスクトップは Access Manager サービスのトップで、ディスプレイプロファイルデータ保管の仕組みを実行します。ディスプレイプロファイルとその他のデスクトップサービスのデータは、管理コンソールで編集できます。デスクトップとディスプレイプロファイルの管理の詳細については、[第 8 章「ポータルデスクトップサービスの管理」](#) および [第 10 章「ディスプレイプロファイルの管理」](#) を参照してください。

リライタ

リライタは、HTML、JavaScript™、および WML などの各種 Web 言語形式、および HTTP ロケーションヘッダー (リダイレクト) 形式の、URL 参照を書き換えるための Java クラスライブラリを提供します。書き換え方法と書き換えるデータを定義するルールを保存するために、リライタは Access Manager サービスを定義します。リライタルールは、管理コンソールで編集できます。リライタの管理については、[第 12 章「リライタサービスの管理」](#) を参照してください。

検索エンジン

検索エンジンサービスは、デスクトップに基本および詳細検索チャンネルとブラウザチャンネルを提供します。検索エンジンサービスは、イントラネットで利用できるドキュメントのリソース記述を作成するためにロボットを使用し、これらのリソース記述をインデックス付きデータベースに格納します。リソース記述 (RD) は、他のサーバーまたはバックアップ SOIF (Summary Object Interchange Format) ファイルからインポートすることもできます。検索エンジンには、リソース記述を送信したり、データベースを検索したりするための Java および C の API が含まれています。検索エンジンデータベースは、他のコンテンツプロバイダのための共有コンテンツキャッシュのように、その他の任意のコンテンツを格納する場合にも使用できます。検索エンジンサービスのデータは、管理コンソールで編集できます。検索の管理については、[第 13 章「検索エンジンサービスの管理」](#) を参照してください。

NetMail

NetMail サービスは、電子メールクライアントの NetMail (Java) および NetMail Lite を実装しています。これらのクライアントは、標準の IMAP および SMTP サーバー環境で動作します。NetMail サービスのデータは、管理コンソールで編集できます。NetMail の管理については、[第 11 章「NetMail サービスの管理」](#) を参照してください。

WSRP

WSRP (Web Server Remote Portlets) を使用すると、リモートアプリケーションおよびコンテンツのポータルへの統合が簡単にできます。WSRP プロデューサと WSRP コンシューマは、管理コンソールを通じて管理できます。WSRP の管理については、[第 9 章「Web Services for Remote Portlets \(WSRP\) サービスの管理」](#) を参照してください。

SSO アダプタ

シングルサインオン (SSO) アダプタサービスを使用することで、エンドユーザーは Portal Server プロバイダやその他の Web アプリケーションなどのアプリケーションを使用し、1 回のサインオンだけで各種リソースサーバーに認証アクセスすることができます。アドレス帳、メール、およびカレンダーサーバーにおける SSO の使用については、[付録 A「SSO アダプタテンプレートと設定」](#) を参照してください。

登録

登録サービスを利用することで、ユーザーは多くの情報を網羅する興味対象のプロファイルを作成できます。詳細については、[第 16 章「登録サービスの管理」](#) を参照してください。

Portal Server サービスのための設定の仕組み

Portal Server はさまざまな設定の仕組みを使用して、サービスを定義、保存、管理します。ここでは5つの表で、Portal Server の各内部サービスで使用される設定の仕組みを説明します。

40 ページの表 1-1 は、デスクトップサービスの設定の仕組みを示しています。この表には、「設定の仕組み」と「説明」という2つの列があります。「設定の仕組み」はメカニズムを、「説明」はメカニズムの目的を示します。

表 1-1 Portal Server デスクトップ設定の仕組み

設定の仕組み	説明
デスクトップサービスの定義	デスクトップサービスの Access Manager の設定属性を定義します。詳しい XML 参照情報については、『Sun Java System Portal Server 2005Q1 Technical Service Guide』を参照。
デスクトップディスプレイプロファイル XML DTD	プロバイダとチャネルオブジェクト、およびそれらのプロパティを定義することで、デスクトップの表示設定を定義します。詳しい XML 参照情報については、『Sun Java System Portal Server 2005Q1 Technical Service Guide』を参照。
デスクトップ管理コンソールモジュール	Portal Server サービスを Access Manager フレームワークで管理するための手段を提供します。デスクトップサービスの設定属性の管理については、第 8 章「ポータルデスクトップサービスの管理」を参照。ディスプレイプロファイルの管理については、第 10 章「ディスプレイプロファイルの管理」を参照。
デスクトップ CLI	製品管理のための dpadmin および par コマンドユーティリティを提供します。コマンド行ユーティリティの管理についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照。
デスクトップ設定プロパティファイル	デスクトップサービスのサーバー固有パラメータを定義します。検索設定プロパティについての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照。

表 1-2 は、検索サービスの設定の仕組みを示しています。この表には、「設定の仕組み」と「説明」という2つの列があります。「設定の仕組み」はメカニズムを、「説明」はメカニズムの目的を示します。

表 1-2 Portal Server 検索設定の仕組み

設定の仕組み	説明
検索サービス定義	検索サービスのための Access Manager の設定属性を定義します。スキーマ参照についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照。
検索管理コンソールモジュール	Portal Server 検索サービスのデータを Access Manager フレームワークで管理するための手段を提供します。詳細については、 第 13 章「検索エンジンサービスの管理」 を参照してください。
検索 CLI	製品管理のための rdmgr、sendrdm、および StartRobot コマンドユーティリティを提供します。コマンド行ユーティリティの管理についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照。
検索設定プロパティファイル	検索サービスのサーバー固有パラメータを定義します。検索設定プロパティについての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照。
ロボット設定ファイル	検索エンジンロボットの動作を定義します。ロボット設定ファイルには 4 種類あります。詳細については、 第 14 章「検索エンジンロボットの管理」 および 第 15 章「定義済みのロボットアプリケーション関数」 を参照。

表 1-3 は、リライターサービスの設定の仕組みを示しています。この表には、「設定の仕組み」と「説明」という 2 つの列があります。「設定の仕組み」はメカニズムを、「説明」はメカニズムの目的を示します。

表 1-3 Portal Server リライター設定の仕組み

設定の仕組み	説明
リライターサービス定義	リライターサービスのための Access Manager の設定属性を定義します。スキーマ参照についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照。
リライタルール XML DTD	詳しい XML 参照情報については、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照。
リライター管理コンソールモジュール	Portal Server リライターサービスのデータを Access Manager フレームワークで管理するための手段を提供します。詳細については、 第 12 章「リライターサービスの管理」 を参照してください。
リライター CLI	製品管理のための radmin コマンドユーティリティを提供します。コマンド行ユーティリティの管理についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照。

表 1-4 は、NetMail サービスの設定の仕組みを示しています。この表には、「設定の仕組み」と「説明」という 2 つの列があります。「設定の仕組み」はメカニズムを、「説明」はメカニズムの目的を示します。

表 1-4 Portal Server NetMail 設定の仕組み

設定の仕組み	説明
NetMail サービス定義	NetMail サービスのための Access Manager の設定属性を定義します。スキーマ参照についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照。
NetMail 管理コンソールモジュール	Portal Server NetMail サービスのデータを Access Manager フレームワークで管理するための手段を提供します。詳細については、第 11 章「NetMail サービスの管理」を参照してください。

管理の概要

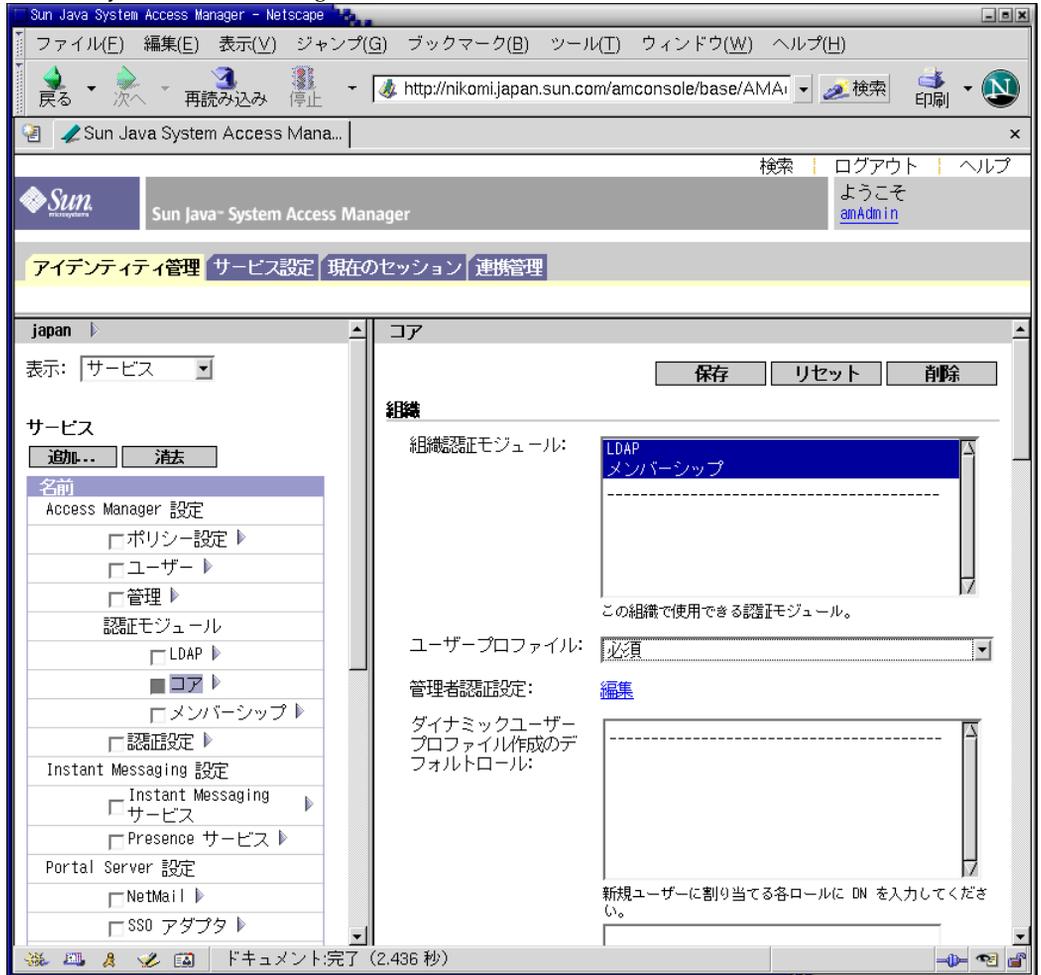
ここでは、Access Manager コンソールとコマンド行の両方から行う Portal Server の管理の概要について説明します。

Access Manager コンソールの使用

Portal Server と Access Manager のサービスの管理には、Access Manager が提供する HTML ベースの管理コンソールを使用します。Portal Server は、Access Manager コンソールを拡張し、Portal Server に固有のサービスの管理モジュールを追加します。コンソールを使用して実際に行う作業の詳細については、本書の各章を参照してください。

Access Manager コンソールは、3 つのセクションに分割されています。ロケーションパネル、ナビゲーションパネル、およびデータパネルです。管理者はこの 3 つのパネルを使用して、ディレクトリの切り替え、ユーザーとサービスの設定、およびポリシーの作成を行います。43 ページの図 1-2 は、管理コンソールを示しています。

図 1-2 Sun Java System Access Manager 管理コンソール



ロケーションパネル

ロケーションパネルは、コンソールの上部に表示されます。最上部のメニューを使用すると、管理者は次の4つの管理画面を切り替えることができます。

- アイデンティティ管理
- サービス設定
- 現在のセッション
- 連携管理

「ようこそ」フィールドには、現在コンソールを実行しているユーザー名とそのユーザープロファイルへのリンクが表示されます。

「ヘルプ」リンクは、『Attribute Reference Guide』の付録 C、D、E、F の HTML バージョンを表示するためのブラウザウィンドウを開きます。

「ログアウト」リンクを使用すると、ユーザーは Access Manager コンソールからログアウトできます。

ナビゲーションパネル

ナビゲーションパネルは、コンソールの左側部分です。パネルの上部にディレクトリオブジェクトの区画があり、現在開いているディレクトリオブジェクトの名前とそのプロパティへのリンクが表示されます。「表示」メニューは、選択したディレクトリオブジェクト内のディレクトリをリストします。サブディレクトリの数により、ページメカニズムが機能します。

データパネル

データパネルは、コンソールの右側部分です。オブジェクト属性とその値は、ここで表示、設定されます。それぞれのグループ、ロール、組織のエントリは、このパネルで選択されます。

コマンドラインユーティリティの使用

Portal Server コマンド行インタフェースは、Access Manager と Portal Server で提供されるユーティリティで構成されます。

Portal Server コマンド行ユーティリティの完全なリストと構文については、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

Access Manager 製品のコマンド行ユーティリティについては、同製品のマニュアルを参照してください。

Sun Java System Portal Server の設定

第 2 章 「インストール後の設定」

第 3 章 「サーバーのインスタンスの作成および削除」

第 4 章 「セキュリティ保護された外部 LDAP Directory Server を
使用する場合の Portal Server の設定」

第 5 章 「root 以外のユーザーとして Portal Server を実行するた
めの設定」

インストール後の設定

Portal Server 設定プログラム

Sun Java™ Enterprise System インストーラでインストール中に、あとで設定することを選択する場合、Portal Server 設定プログラムを使って Portal Server コンポーネント製品を設定します。次のチェックリストは、Portal Server コンポーネント製品の設定に使用するパラメータの説明です。

設定プログラムの実行

設定プログラムを実行するには、次の手順に従います。

1. ターミナルウィンドウで `root` として、設定プログラムを含む次のディレクトリに移動します。

```
cd PortalServer-base/SUNWps/lib
```

2. 次のように入力し、設定プログラムスクリプトを実行します。

```
./configurator
```

注 デバッグを有効にするには、次のように入力します。

```
configurator -DPS_CONFIG_DEBUG=y
```

デバッグを有効にすると、デバッグ情報と同時にパスワードが画面に表示されます。

3. 設定画面の指示に従ってください。

ローカライズ環境での設定プログラムの実行

ローカライズ環境では、特定の領域または指定された言語の動作環境で機能するように、ソフトウェア製品が変更されています。

Solaris Sparc または x86 コンピュータ上のローカライズ環境で設定プログラムを実行するには、次を 1 行で入力します。

```
/usr/jdk/entsys-j2se/bin/java -DPS_CONFIG_DEBUG=y
-DDEPLOY_ADMIN_PASSWORD=<deploy admin password>
-DDS_DIRMGR_PASSWORD=<LDAP directory manager password>
-DIDSAME_LDAPUSER_PASSWORD=<AM ldap user password>
-DIDSAME_ADMIN_PASSWORD=<AM admin password>
-DBASEDIR=PS-INSTALL-DIR
-cp PS-INSTALL-DIR/SUNWps/lib/configL10N.jar
com.sun.portal.config.ConfigureL10N
```

Secure Remote Access で L10N 設定プログラムを実行するには、次のようにコマンドを入力します。

```
/usr/jdk/entsys-j2se/bin/java -DPS_CONFIG_DEBUG=y
-DDEPLOY_ADMIN_PASSWORD=<deploy admin password>
-DDS_DIRMGR_PASSWORD=<LDAP directory manager password>
-DIDSAME_LDAPUSER_PASSWORD=<AM ldap user password>
-DIDSAME_ADMIN_PASSWORD=<AM admin password>
-DBASEDIR=PS-INSTALL-DIR
-cp PS-INSTALL-DIR/SUNWps/lib/configL10N.jar
com.sun.portal.config.ConfigureL10N
```

Linux コンピュータ上のローカライズ環境で設定プログラムを実行するには、次を 1 行で入力します。

```
/usr/jdk/entsys-j2se/bin/java -DPS_CONFIG_DEBUG=y
-DDEPLOY_ADMIN_PASSWORD=<deploy admin password>
-DDS_DIRMGR_PASSWORD=<LDAP directory manager password>
-DIDSAME_LDAPUSER_PASSWORD=<AM ldap user password>
-DIDSAME_ADMIN_PASSWORD=<AM admin password>
-DBASEDIR=PS-INSTALL-DIR
```

```
-cp PS-INSTALL-DIR/sun/portal/lib/configL10N.jar
com.sun.portal.config.ConfigureL10N
```

SRA で L10n 設定プログラムを実行するには、次のようにコマンドを入力します。

```
/usr/jdk/entsys-j2se/bin/java -DPS_CONFIG_DEBUG=y
-DDEPLOY_ADMIN_PASSWORD=<deploy admin password>
-DDS_DIRMGR_PASSWORD=<LDAP directory manager password>
-DIDSAME_LDAPUSER_PASSWORD=<AM ldap user password>
-DIDSAME_ADMIN_PASSWORD=<AM admin password>
-DBASEDIR=PS-INSTALL-DIR
-cp PS-INSTALL-DIR/sun/portal/lib/configL10N.jar
com.sun.portal.config.ConfigureL10N SRA
```

設定チェックリスト

インストール中に後で設定することを選択した場合は、Sun Java™ System Portal Server 設定プログラムを使用して、Portal Server インストールを設定する必要があります。次のチェックリストは、インストール後の設定に必要な値の説明です。実行するインストールの種類によって、使用する値は異なる場合があります。

Portal Server と Secure Remote Access

表 2-1 には 3 つの列があり、インストール中に後で設定することを選択した場合に必要なすべての値を示しています。実行するインストールの種類によって、使用する値は異なる場合があります。

表 2-1 は、Sun Java System Application Server に配備を想定したチェックリストの例です。Portal Server を、Sun Java System Web Server、BEA WebLogic、または IBM WebSphere Application Server に配備する場合、それらの Web コンテナ値については「[Web コンテナのチェックリスト](#)」を参照してください。

表 2-1 Portal Server のインストールチェックリスト

パラメータ	デフォルト値	説明
インストールディレクトリ		
Sun Java System 設定プログラムコンポーネントをインストールするディレクトリ	/opt	Portal Server ソフトウェアがインストールされているベースディレクトリです。
配備情報		
配備タイプ	Sun Java System Application Server	Portal Server は、Sun Java System Web Server、Sun Java System Application Server、BEA WebLogic Server、または IBM WebSphere Application Server への配備が可能です。
配備 URI	/portal	<p>配備 URI は、Portal Server が使用する Web サーバーまたはアプリケーションサーバーの領域です。</p> <p>配備 URI の値には、先頭にスラッシュを付けますが、使用できるスラッシュはこれだけです。また、配備 URI を「/」のみにすることはできません。</p>
Portal Server インスタンスを制御するロードバランサ	チェックボックスがオフ	Portal Server と共にロードバランサを使用する場合は、このボックスにチェックマークを付けます。
ロードバランサの URL	http://fully_qualified_domain:port/portal	
Web コンテナ情報 (Sun Java System Application Server)		
インストールディレクトリ	/opt/SUNWappserver/appserver	Sun Java System Application Server がインストールされているディレクトリ。
ドメイン	/var/opt/SUNWappserver7/domains/domain1	Sun Java System Application Server ドメインには、一連のインスタンスが含まれます。指定するドメインには、Portal Server に使用されるインスタンスが含まれます。このドメインは設定済みにする必要があります。
インスタンス	server	<p>Portal Server が配備される Sun Java System Application Server インスタンスの名前。このインスタンスは設定済みにする必要があります。</p> <p>インスタンス名にはスペースを含めてはいけません。</p>

表 2-1 Portal Server のインストールチェックリスト (続き)

パラメータ	デフォルト値	説明
インスタンスポート	8080	Sun Java System Application Server インスタンスが動作するポート。
ドキュメントルートディレクトリ	/var/opt/SUNWappserver/domains/domain1/docroot	スタティックなページが格納されるディレクトリ。
管理者	admin	管理者のユーザー ID。
管理ポート	4849	管理サーバーのポート番号。
管理パスワード		管理サーバーのパスワード。
Access Manager 情報		
インストールベースディレクトリ	/opt	Sun Java System Access Manager ソフトウェアがインストールされているベースディレクトリ。
内部 LDAP 認証ユーザーパスワード		Sun Java System Access Manager インストール時に選択された内部 LDAP 認証ユーザーパスワード。
管理 (amadmin) パスワード		Sun Java System Access Manager ソフトウェアのインストール時に選択されたトップレベルの管理 (amadmin) パスワード。
Directory Manager の DN	cn=Directory Manager	LDAP ディレクトリマネージャの識別名 (DN)。
Directory Manager のパスワード		Sun Java System Directory Server のインストール時に選択されたディレクトリマネージャのパスワード。
Secure Remote Access 情報 (Secure Remote Access サポートの設定用)		
ゲートウェイプロトコル	https	ゲートウェイが使用するプロトコル。ゲートウェイは Secure Sockets Layer (SSL) を使用して通信を行います。
Portal Server のドメイン	portal-server-domain-name	Portal Server をインストールするコンピュータのドメイン名。
ゲートウェイのドメイン	gateway-domain-name	ゲートウェイコンピュータのドメイン名。
ゲートウェイポート	443	ゲートウェイが待機するポート。

表 2-1 Portal Server のインストールチェックリスト (続き)

パラメータ	デフォルト値	説明
ゲートウェイプロファイル名	default	ゲートウェイプロファイルには、ゲートウェイが待機するポート、SSL オプション、プロキシオプションなど、ゲートウェイの構成に関連するすべての情報が記録されています。 ゲートウェイ管理コンソールで複数のプロファイルを作成し、ゲートウェイのインスタンスごとに別のプロファイルに関連付けることができます。 『Sun Java System Portal Server, Secure Remote Access 6 2005Q1 管理ガイド』の「ゲートウェイプロファイルの作成」を参照。
ゲートウェイのログイン ユーザーパスワード		root 以外の管理者に、ゲートウェイのログファイルを参照する権限を付与します。
パスワードの確認		確認のためにパスワードを再入力します。

ゲートウェイ

表 2-2 ゲートウェイのインストールチェックリスト

パラメータ	デフォルト値	説明
プロトコル	https	ゲートウェイが使用するプロトコル。ゲートウェイは通常、Secure Sockets Layer (SSL) を使用して通信を行います。
ホスト名	host	ゲートウェイをインストールするコンピュータのホスト名。
サブドメイン ドメイン	gateway-subdomain-name gateway-domain-name	ゲートウェイコンピュータのサブドメイン名。 ゲートウェイコンピュータのドメイン名。
IP アドレス	host-ip-address	Sun Java System Access Manager の IP アドレスではなく、ゲートウェイをインストールするコンピュータの IP アドレス。
アクセスポート	443	ゲートウェイコンピュータが待機するポート。

表 2-2 ゲートウェイのインストールチェックリスト (続き)

パラメータ	デフォルト値	説明
ゲートウェイプロファイル名	default	Portal Server または Secure Remote Access サポートのインストール時に指定したプロファイル名を指定します。 詳細については、『Sun Java System Portal Server, Secure Remote Access 6 2005Q1 管理ガイド』の「ゲートウェイプロファイルの作成」を参照。
ログユーザーパスワード		root 以外の管理者に、ゲートウェイのログファイルを参照する権限を付与します。
インストール後にゲートウェイを起動する	Selected	インストール後にゲートウェイを自動的に起動するか (このオプションをオンにした場合)、あるいは後で起動するかを指定します。 ゲートウェイを手動で起動するには、 <i>PortalServer-base/SUNWps/bin</i> に含まれる次のコマンドを使用します。 <code>./gateway -n gateway-profile-name start</code>
証明書情報		
組織	MyOrganization	組織の名前。
部門	MyDivision	部門の名前。
都市または地域	MyCity	都市または地域の名前。
都道府県	MyState	都道府県名。
2 文字の国コード	us	自国の 2 文字の国コード。
証明書のデータベースのパスワード		任意のパスワードを指定できます。
パスワードの再入力		確認のためにパスワードを再入力します。

Netlet プロキシ

表 2-3 Netlet プロキシのインストールチェックリスト

パラメータ	デフォルト値	説明
ホスト名	<i>hostname</i>	Netlet プロキシをインストールするコンピュータのホスト名。
サブドメイン	<i>localhost-subdomain-name</i>	Netlet プロキシをインストールするコンピュータのサブドメイン名。
ドメイン	<i>localhost- domain-name</i>	Netlet プロキシをインストールするコンピュータのドメイン名。
IP アドレス	<i>host-ip-address</i>	Sun Java System Access Manager の IP アドレスではなく、Netlet プロキシをインストールするコンピュータの IP アドレス。
アクセスポート	10555	Netlet プロキシが待機するポート。
ゲートウェイプロファイル名	default	Portal Server または Secure Remote Access サポートのインストール時に指定したプロファイル名を指定します。 詳細については、『Sun Java System Portal Server, Secure Remote Access 6 2005Q1 管理ガイド』の「ゲートウェイプロファイルの作成」を参照。
ゲートウェイのログインユーザーパスワード		root 以外の管理者に、ゲートウェイのログファイルを参照する権限を付与します。
インストール後に Netlet プロキシを起動する	Selected	インストール後に Netlet プロキシを自動的に起動するか (このオプションをオンにした場合)、あるいは後で起動するかを指定します。Netlet プロキシを手動で起動するには、 <i>NetletProxy-base/SUNWps/bin</i> に含まれる次のコマンドを使用します。 <code>./netletd -n default start</code>
証明書情報		
組織	MyOrganization	組織の名前。
部門	MyDivision	部門の名前。
都市または地域	MyCity	都市または地域の名前。
都道府県	MyState	都道府県名。
2 文字の国コード	us	自国の 2 文字の国コード。

表 2-3 Netlet プロキシのインストールチェックリスト (続き)

パラメータ	デフォルト値	説明
証明書のデータベースのパスワード		任意のパスワードを指定できます。
パスワードの再入力		確認のためにパスワードを再入力します。

リライタプロキシ

表 2-4 リライタプロキシのインストールチェックリスト

パラメータ	デフォルト値	説明
ホスト名	<i>hostname</i>	リライタプロキシをインストールするコンピュータのホスト名。
サブドメイン	<i>localhost-subdomain-name</i>	リライタプロキシをインストールするコンピュータのサブドメイン名。
ドメイン	<i>localhost- domain-name</i>	リライタプロキシをインストールするコンピュータのドメイン名。
IP アドレス	<i>host-ip-address</i>	Sun Java System Access Manager の IP アドレスではなく、リライタプロキシをインストールするコンピュータの IP アドレス。
アクセスポート	10443	リライタプロキシが待機するポート。
ゲートウェイプロファイル名	default	Portal Server または Secure Remote Access サポートのインストール時に指定したプロファイル名を指定します。 詳細については、『Sun Java System Portal Server, Secure Remote Access 6 2005Q1 管理ガイド』の「ゲートウェイプロファイルの作成」を参照
ゲートウェイのログインユーザーパスワード		root 以外の管理者に、ゲートウェイのログファイルを参照する権限を付与します。
インストール後にリライタプロキシを起動する	Selected	インストール後にリライタプロキシを自動的に起動するか (このオプションをオンにした場合)、あるいは後に手動で起動するかを指定します。 リライタプロキシを手動で起動するには、 <i>RewriterProxy-base/SUNWps/bin</i> に含まれる次のコマンドを使用します。 <code>./rwproxyd -n default start</code>

証明書情報

表 2-4 リライタプロキシのインストールチェックリスト (続き)

パラメータ	デフォルト値	説明
組織	MyOrganization	組織の名前。
部門	MyDivision	部門の名前。
都市または地域	MyCity	都市または地域の名前。
都道府県	MyState	都道府県名。
2 文字の国コード	us	自国の 2 文字の国コード。
証明書のデータベースのパスワード		任意のパスワードを指定できます。
パスワードの確認		確認のためにパスワードを再入力します。

表 2-5 Netlet プロキシとリライタプロキシの情報

パラメータ	デフォルト値	説明
このノードの Portal Server を使用する	Selected	Portal server と Sun Java System Secure Remote Access コンポーネントを異なるコンピュータにインストールする場合は、このボックスのチェックマークを外します。 Portal server とその他の Sun Java System Secure Remote Access コンポーネントを同じコンピュータにインストールする場合は、このボックスにチェックマークを付けます。 このボックスにチェックマークを付けた場合、次のフィールドが有効になります。
Portal Server のプロトコル	http	Portal Server が通信に使用するプロトコル。
Portal Server のホスト	<i>fully-qualified-host-name</i>	Portal Server の完全修飾ホスト名。
Portal Server のポート	80	
Portal Server の配備 URI	/portal	
組織 DN		
Access Manager のサービス URI	/amserver	

表 2-5 Netlet プロキシとリライタプロキシの情報

パラメータ	デフォルト値	説明
Access Manager パスワード 暗号化鍵		暗号化鍵の値。暗号化鍵は次の場所にありません。 /etc/opt/SUNWam/config AMConfig.properties: この値を含むパラメータは am.encryption.pwd です。

Web コンテナのチェックリスト

Portal Server のインストールは、一部の Web コンテナのパラメータに依存します。次のチェックリストは、Portal Server のインストール処理中に必要となるパラメータを示しています。Portal Server 製品を配備している Web コンテナに応じたチェックリストを参照してください。

- [Sun Java System Web Server のチェックリスト](#)
- [Sun Java System Application Server のチェックリスト](#)
- [BEA WebLogic Server のチェックリスト](#)
- [IBM WebSphere Application Server のチェックリスト](#)

Portal Server でサポートされるアプリケーションサーバーの使用の詳細については、このマニュアルの付録の、使用するアプリケーションサーバーの配備に対応する箇所を参照してください。

Sun Java System Web Server のチェックリスト

表 2-6 Portal Server のインストール中に使用される Sun Java System Web Server の値

パラメータ	デフォルト値	説明
インストールディレクトリ	/opt/SUNWwbsvr	Sun Java System Web Server がインストールされているベースディレクトリ。
インスタンス	host	Portal Server に使用させる Web サーバーインスタンス。 インスタンス名にはスペースを含めてはいけません。
インスタンスポート	80	Portal Server へのアクセスに使用するポート。
セキュア Web コンテナのインスタンスポート	チェックボックスがオフ	インスタンスポート上で SSL を実行する場合は、このボックスにチェックマークを付けます。
ドキュメントルートディレクトリ	/opt/SUNWwbsvr/docs	スタティクなページが格納されるディレクトリ。

Sun Java System Application Server のチェックリスト

表 2-7 Portal Server のインストール中に使用される Sun Java System Application Server の値

パラメータ	デフォルト値	説明
インストールディレクトリ	/opt/SUNWappserver/appserver	Sun Java System Application Server がインストールされているディレクトリ。
ドメイン	/var/opt/SUNWappserver/domains/domain1	Sun Java System Application Server ドメインには、一連のインスタンスが含まれます。指定するドメインには、Portal Server に使用されるインスタンスが含まれます。このドメインは設定済みにする必要があります。
インスタンス	server	Portal Server が配備される Sun Java System Application Server インスタンスの名前。このインスタンスは設定済みにする必要があります。 インスタンス名にはスペースを含めてはいけません。
インスタンスポート	8080	Portal Server へのアクセスに使用するポート。

表 2-7 Portal Server のインストール中に使用される Sun Java System Application Server の値 (続き)

パラメータ	デフォルト値	説明
セキュア Web コンテナのインスタンスポート	チェックボックスがオフ	インスタンスポート上で SSL を実行する場合は、このボックスにチェックマークを付けます。
ドキュメントルートディレクトリ	/var/opt/SUNWappserver/domains/domain1/docroot	スタティックなページが格納されるディレクトリ。
管理者	admin	管理者のユーザー ID。
管理ポート	4849	管理サーバーのポート番号。
管理パスワード		管理サーバーのパスワード。

BEA WebLogic Server のチェックリスト

表 2-8 Portal Server のインストール中に使用される BEA WebLogic Server の値

パラメータ	デフォルト値	説明
BEA ホームディレクトリ	/user/local/boa	BEA がインストールされているディレクトリ。
製品インストールディレクトリ	/usr/local/boa/weblogic81	BEA WebLogic Server ソフトウェアがインストールされているディレクトリ。
ユーザープロジェクトのディレクトリ	user_projects	BEA WebLogic のインストール中に入力した値を使用します。
ドメイン	mydomain	BEA WebLogic Server ドメインには、一連のインスタンスが含まれます。指定するドメインには、Portal Server に使用されるインスタンスが含まれます。このドメインは設定済みにする必要があります。
インスタンス	myserver	Portal Server が配備される BEA WebLogic Server インスタンスの名前。このインスタンスは設定済みにする必要があります。 インスタンス名にはスペースを含めてはいけません。 Portal Server を管理サーバーインスタンス上にインストールする場合は、管理サーバーインスタンスの名前が指定されます。そうでない場合は、管理されているサーバーインスタンスの名前が指定されます。

表 2-8 Portal Server のインストール中に使用される BEA WebLogic Server の値 (続き)

パラメータ	デフォルト値	説明
インスタンスポート	7001	Portal Server へのアクセスに使用するポート。
セキュアインスタンスポート	チェックボックスがオフ	インスタンスポート上で SSL を実行する場合は、このボックスにチェックマークを付けます。
ドキュメントルートディレクトリ	/usr/local/bea/user_projects/domains/mydomain/applications	DefaultWebApp のドキュメントルート値は、Portal Server ソフトウェアを実行している BEA WebLogic Server インスタンスに配備されている必要があります。DefaultWebApp はデフォルトの Web アプリケーションであり、BEA WebLogic Server のスタティックコンテンツはここから提供されます。デフォルトでは、DefaultWebApp はドメイン (mydomain) と、BEA WebLogic Server のインストール中に定義または作成されるサーバーインスタンスだけに配備されます。したがって、独自の BEA WebLogic Server またはドメインを作成する場合は、ディレクトリを新しいサーバーの配備ディレクトリにコピーするか、または BEA WebLogic Server 管理コンソールを使用して、DefaultWebApp をそこに配備する必要があります。デフォルトの Web アプリケーションを構成する方法の詳細については、BEA WebLogic Server のマニュアルを参照。
管理者	system	管理者のユーザー ID。
管理プロトコル	http	BEA WebLogic Server の管理サーバーが使用するプロトコル。
管理ポート	7001	BEA WebLogic Server の管理サーバーが動作しているポート。Portal Server が BEA WebLogic Server 管理サーバー自体にインストールされている場合、Portal Server が動作するポートと BEA WebLogic Server の管理ポートは同じです。
管理パスワード		システムのパスワード。

IBM WebSphere Application Server のチェックリスト

表 2-9 Portal Server のインストール中に使用される IBM WebSphere Application Server の値

パラメータ	デフォルト値	説明
インストールディレクトリ	/opt/WebSphere/Express51 /AppServer	IBM WebSphere Application Server ソフトウェアがインストールされているディレクトリ。
仮想ホスト	default_host	IBM WebSphere のインストール中に入力した値を使用します。
ノード	machine-name	
インスタンス	server1	Portal Server が配備されるインスタンスの名前。このインスタンスは設定済みにする必要があります。
ドキュメントルートディレクトリ	/opt/IBMHTTPD/htdocs/en_US	Portal Server は、Default-Server または Default Server のように、名前にダッシュやスペースを含むアプリケーションサーバーインスタンスやドメインにインストールすることはできません。 スタティックなページが格納されるディレクトリ。このディレクトリは Sun Java System Access Manager のインストール中に作成されます。

Portal Server インストール後の作業

インストール後の作業は、次のコンポーネントごとに実行する必要があります。

- [Portal Server](#)
- [Secure Remote Access](#)
- [ゲートウェイ](#)
- [Netlet プロキシとリライタプロキシ](#)

Portal Server

Portal Server または Access Manager の管理コンソールにアクセスするには、ディレクトリサーバーと Web コンテナを先に起動しておく必要があります。

次のコマンドを使用して、ディレクトリサーバーのローカルインストールを開始します。

```
/var/opt/mps/serverroot/slapd-hostname/start-slapd
```

次のインストール後作業は、Portal Server を配備している Web コンテナの種類によって異なります。

- [Sun Java System Web Server](#)
- [Sun Java System Application Server](#)
- [BEA WebLogic Server](#)
- [IBM WebSphere Application Server](#)

Sun Java System Web Server

Sun Java System Web Server を起動するには、次の手順に従います。

1. `admin` インスタンスを起動します。ターミナルウィンドウで、次のように入力します。

```
cd WebServer-base/SUNWwbsrv/https-server-instance-name  
./start
```

2. Sun Java System Web Server 管理コンソールにアクセスします。
3. 「Apply Changes」をクリックして、Web コンテナを再起動します。

Sun Java System Application Server

アプリケーションサーバーインスタンスの設定

1. ドメインを起動します。ターミナルウィンドウで、次のように入力します。

```
cd $AS8.1_INSTALLDIR/appserver/bin
```

1行で次のように入力します。

```
./asadmin start-domain --user admin --passwordfile passwordfilename domain1
```

ドメインを起動すると、**admin** サーバーおよびほかのすべてのインスタンスが起動します。

2. 特定のインスタンスを起動するには、ターミナルウィンドウで次のように入力します。

```
cd $AS8.1_INSTALLDIR/appserver/bin
```

1行で次のように入力します。

```
./asadmin start-instance --user admin_user [--passwordfile filename] [--host host_name] [--port port_number] instance_name
```

3. 特定のインスタンスを停止するには、ターミナルウィンドウで次のように入力します。

```
cd $AS8.1_INSTALLDIR/appserver/bin
```

1行で次のように入力します。

```
./asadmin stop-instance --user admin_user [--passwordfile filename] [--host host_name] [--port port_number] instance_name
```

Sun Java System Application Server の起動と停止

Sun Java System Application Server インスタンスを起動します。

ターミナルウィンドウで、ディレクトリをアプリケーションサーバーのインスタンスユーティリティディレクトリに変更し、startserv スクリプトを実行します。次の例は、デフォルトのアプリケーションサーバードメインおよびインスタンスを使用した場合を想定したものです。

```
cd /var/opt/SUNWappserver7/domains/domain1/server1/bin
```

```
./startserv
```

asadmin ユーティリティを使用して、または Sun Java System Application Server 管理コンソールから Sun Java System Application Server を起動および停止する方法については、Sun Java System Application Server のマニュアルを参照してください。

BEA WebLogic Server

Portal Server を BEA WebLogic Server 上に配備する場合は、Portal Server ソフトウェアのインストール後に次の手順を実行します。

1. `/var/sadm/install/logs/Java_Enterprise_System_install.B/MMddhhmm` ファイルにエラーがないか確認します。

MM = 月

dd = 日

hh = 時

mm = 分

2. すべての BEA WebLogic Server インスタンスを停止します (admin および managed サーバー)。
3. BEA WebLogic admin サーバーインスタンスを起動します。Portal Server を管理インスタンスにインストールしている場合は、管理インスタンスも起動する必要があります。
4. コマンドラインから、次のコマンドを実行します。

```
PortalServer-base/SUNWps/bin/deploy
```

配備 URI とサーバーインスタンス名にデフォルト値を指定し、パスワード入力を要求されたら、BEA WebLogic Server admin パスワードを入力します。

5. 次のコマンドを実行します。
 - a. `setenv DEPLOY_ADMIN_PASSWORD BEA-WebLogic-admin-password`
 - b. `setenv IDSAME_ADMIN_PASSWORD Identity-Server-admin-password`
 - a. `PortalServer-base/SUNWps/lib/postinstall_PortletSamples`

パスワード入力を要求されたら、BEA WebLogic Server 管理パスワードと Access Manager 管理パスワードを入力します。

これで、`portletsamples.war` ファイルが配備されます。

6. Portal Server が配備されている BEA WebLogic Server インスタンスを再起動します。Web コンテナのインスタンスを起動する手順については、使用する Web コンテナのマニュアルを参照してください。

注 管理サーバーをインストールする場合、`.war` ファイルを配備することはありません。`.war` ファイルは、BEA WebLogic Server 管理コンソールを使用して配備する必要があります。

LDAP、UNIX、匿名など複数の認証方法をサポートする場合は、それぞれの認証タイプをコア認証サービスに追加して、認証メニューを作成する必要があります。詳細については『Sun Java System Portal Server 6 2004Q2 管理ガイド』を参照してください。

IBM WebSphere Application Server

1. `/var/sadm/install/logs/Java_Enterprise_System_install.B/MMddhhmm` ファイルにエラーがないか確認します。
2. アプリケーションサーバーインスタンスとアプリケーションサーバーノードを停止して再起動します。Web コンテナインスタンスの起動手順については、使用する Web コンテナのマニュアルを参照してください。

Secure Remote Access

Portal Server をゲートウェイと共に使用する場合、ゲートウェイ認証局 (CA) の証明書を Portal Server の信頼できる CA リストに追加する必要があります。この場合、Portal Server が HTTP と HTTPS どちらのモードで実行されているかは関係ありません。

ユーザーセッションのタイムアウトまたはログアウトが発生した場合、Sun Java System Access Manager はゲートウェイにセッション通知を送信します。Sun Java System Access Manager が HTTP モードで実行中でも、`HttpsURLConnection` を使用してセッション通知を送信することにより、SSL クライアントの役割を果たします。`HttpsURLConnection` は SSL サーバー (ゲートウェイ) に接続しているため、信頼できる CA リストの一部としてゲートウェイ CA 証明書を含めるか、またはオプションで自己署名証明書を有効にする必要があります。

注 信頼できる CA リストに CA を追加する方法は、定義されているプロトコルハンドラによって異なります。

`HttpsURLConnection` を作成するには、Java Virtual Machine (JVM™) プロパティ `-Djava.protocol.handler.pkgs` を設定する必要があります。

Portal Server が、Sun Java System Web Server、Sun Java System Application Server、または BEA WebLogic Server 上で動作している場合、上のプロパティはデフォルトで `com.iplanet.services.comm` に正しく設定されます。Sun Java System Access Manager パッケージには `HttpsURLConnection` の実装が含まれているため、`AMConfig.properties` ファイルにフラグ `com.iplanet.am.jssproxy.trustAllServerCerts=true` を追加することで、あらゆる SSL サーバーの自己署名証明書を受け入れることができます。

-Djava.protocol.handler.pkgs は、IBM WebSphere Application Server のデフォルトでは設定されません。サポートされるアプリケーションサーバーの `HttpsURLConnection` の実装では、それぞれに固有のデフォルトのハンドラ (JSSE またはカスタム SSL が実装されている場合がある) を使用する必要があります。

ゲートウェイ

1. 次のコマンドを使用して、ゲートウェイを起動します。

```
Gateway-base/SUNWps/bin/gateway -n new-profile-name start
```

default は、インストール中に作成されるゲートウェイプロファイルのデフォルト名です。後で自分の固有のプロファイルを作成し、新しいプロファイルでゲートウェイを再起動します。『Sun Java System Portal Server, Secure Remote Access 6 2005Q1 管理ガイド』の第 2 章にある「ゲートウェイプロファイルの作成」を参照してください。

複数のゲートウェイインスタンスがある場合は、次のコマンドを使用します。

```
Gateway-base/SUNWps/bin/gateway start
```

注 ゲートウェイのインストール時に、「インストール後にゲートウェイを起動する」オプションを「**y**」と指定した場合、この手順は不要です。

警告 /etc/opt/SUNWps ディレクトリに、起動するインスタンスの設定ファイルだけが含まれていることを確認してください。

特定のノードで動作しているすべてのゲートウェイインスタンスを停止したい場合は、次のコマンドを使用します。

```
Gateway-base/SUNWps/bin/gateway stop
```

Netlet とゲートウェイは、PAC ファイルをサポートするために Rhino JavaScript™ パーサー (SUNWrhino としてバンドルされている) を必要とします。このパーサーは、ゲートウェイおよび Portal Server ノードにインストールされていなければなりません。インストールするには、次の手順を実行します。

1. 現在のディレクトリから `pkgadd -d . SUNWrhino` を実行して、このパッケージを追加します。
2. `package-base/js.jar` to `${JAVA_HOME}/jre/lib/ext` ディレクトリをコピーします。

Netlet プロキシとリライタプロキシ

Netlet プロキシとリライタプロキシを起動する前に、ゲートウェイプロファイルが新しい Netlet プロキシおよびリライタプロキシのオプションで更新されていることを確認してください。

- インストール時に Netlet プロキシを起動するというオプションを選択しなかった場合、Netlet プロキシを手動で起動できます。ディレクトリ *Portal-proxy-base/SUNWps/bin* で、次のように入力します。

```
./netletd -n default start
```
- インストール時にリライタプロキシを手動で起動するというオプションを選択しなかった場合、リライタプロキシを手動で起動できます。ディレクトリ *Portal-proxy-base/SUNWps/bin* で、次のように入力します。

```
./rwproxyd -n default start
```

注 必ずすべてのユーザーのアクセスリストサービスを有効にして、ゲートウェイを介したアクセスを許可してください。

Sun Java System Portal Server ソフトウェア NetFile は、Microsoft Windows にアクセスするために JCIIFS ライブラリ (SUNWjcifs としてバンドルされている) を必要とします。このライブラリは、Portal Server ノードだけにインストールされていなければなりません。インストールするには、次の手順を実行します。

1. 現在の (この) ディレクトリから `pkgadd -d . SUNWjcifs` を実行して、このパッケージを追加します。
2. `PortalServer-base/SUNWps/lib/postinstall_JCIIFS` を実行します。
3. `PortalServer-base/SUNWps/bin/undeploy` を実行します。
4. `PortalServer-base/SUNWps/bin/deploy` コマンドを実行します。
5. サーバーを再起動します。

Portal Server インストールの確認

Portal Server 管理コンソールおよびデスクトップへのアクセス

Sun Java System Access Manager 管理コンソールにアクセスする

1. ブラウザを開きます。
2. `protocol://hostname.domain:port/amconsole` と入力します。
たとえば次のように指定します。
`http://example.com:80/amconsole`
3. 管理者名とパスワードを入力して、管理コンソールを表示します。
これは、Sun Java System Access Manager ソフトウェアのインストール時に指定した名前とパスワードです。

Portal Server デスクトップにアクセスする

デスクトップにアクセスして、Portal Server のインストールを確認します。次の URL を使用して、デスクトップにアクセスします。

`protocol://fully-qualified-hostname:port/portal-URI`

たとえば次のように指定します。

`http://example.com:80/portal`

デスクトップにアクセスすると、認証なしデスクトップが表示されます。これで、ユーザーはデスクトップの URL にアクセス可能となり、自動的に認証されてデスクトップへのアクセスを許可されます。

ポータルデスクトップのサンプルが例外なく表示された場合、Portal Server は正しくインストールされています。

ゲートウェイのインストールの確認

1. 次のコマンドを実行して、特定のポート上でゲートウェイが動作しているかどうかを確認します。

```
netstat -an | grep port-number
```

デフォルトのゲートウェイポートは 443 です。

ゲートウェイが動作していない場合は、デバッグモードでゲートウェイを起動し、コンソールに出力されるメッセージを確認します。次のコマンドを使用して、ゲートウェイをデバッグモードで起動します。

```
PortalServer-base/SUNWps/bin/gateway -n profilename start debug
```

platform.conf.profilename ファイルの gateway.debug 属性を message に設定して、ログファイルを参照します。詳細については、『Sun Java System Portal Server, Secure Remote Access 6 2005Q1 管理ガイド』の第 2 章「ゲートウェイ」にある「platform.conf ファイルの概要」を参照してください。

2. ブラウザで次のゲートウェイ URL を入力して、Portal Server をセキュアモードで実行します。

```
https://gateway-machine-name:portnumber
```

インストール時にデフォルトポート (443) を選択した場合には、ポート番号を指定する必要はありません。

3. ユーザー名 amadmin とインストール時に指定したパスワードを使用して、Access Manager 管理コンソールに管理者としてログインします。

これで、管理コンソールで新しい組織、ロール、ユーザーを作成し、必要なサービスと属性を割り当てることができます。

ゲートウェイのインストールの確認

サーバーのインスタンスの作成および削除

インスタンスは、1 つ以上の IP アドレスにバインドされた、特定のポートで待機するサーバーです。Sun Java™ System Portal Server の場合、ポートで待機し、単一の Java™ 仮想マシン (JVM™) を実行する Web サーバープロセスに、インスタンスが対応します。

注 Sun Java™ System Web Server および Sun Java™ System Application Server では複数インスタンスがサポートされます。

サーバーのインスタンスを作成する

1. Portal Server が動作しているサーバーにログインします。
2. 新しい Access Manager インスタンスを配備します。新しい Access Manager インスタンスの配備の手順に関しては、次の URL にある『Sun Java System Access Manager 管理ガイド』の第 1 章を参照してください。
<http://docs.sun.com/source/819-1938?l=ja>
3. 次に示す Portal Server のユーティリティディレクトリに移動します。
`cd PortalServer-base/SUNWps/bin`
4. multiserverinstance スクリプトを実行します。
`./multiserverinstance`
5. インスタンス名を入力します。
6. 新規インスタンスのポートを入力します。
7. ポートレットがある場合は、それらのポートレットを再配備します。ポートレットの再配備については、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

8. multiservinstance スクリプトが終了したら、Web サーバーインスタンスのディレクトリに移動します。

```
cd WebServer-base/https-new-instance-name
```

9. Web サーバーインスタンスを停止します。

```
./stop
```

10. Web サーバーインスタンスを再起動します。

```
./start
```

11. ブラウザで新規作成されたインスタンスに移動します。

12. 新規作成されたそれぞれのインスタンスで、手順 9 から手順 11 を繰り返します。

13. ブラウザで次のように入力します。

- 新規インスタンスによって管理コンソールにアクセスする場合は、`http://hostname.domain:instance-portnumber/amconsole` と入力します。
- 新規インスタンスによってポータルデスクトップにアクセスする場合は、`http://hostname.domain:instance-portnumber/portal` と入力します。

追加のサーバーインスタンスを作成し、それらのインスタンスを root 以外または未認証のユーザーとして実行する場合は、

`AccessManager-base/SUNWam/bin/amserver.instance-nickname` で、インスタンスごとに次の行をコメントとして記述します。

```
if [ "$uid" != "0" ];  
then  
    echo "$gettext 'You must be root user to run' '$0.'  
    exit 1  
fi
```

サーバーのインスタンスを削除する

1. Portal Server が動作しているサーバーにログインします。
2. ディレクトリを、`PortalServer-base/SUNWps/bin` に変更します。

```
cd PortalServer-base/SUNWps/bin
```

3. ポートレットがある場合は、それらのポートレットを削除します。コマンド行ユーティリティの管理についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

4. 次のように入力します。

```
./multiserverinstance delete -instance instance-name
```

5. Sun Java System Access Manager も削除する場合は、Access Manager インスタンスをアンインストールします。新しい Access Manager インスタンスのアンインストールの手順に関しては、次の URL にある『Sun Java System Access Manager 管理ガイド』の第 1 章を参照してください。
<http://docs.sun.com/source/819-1938?l=ja>

セキュリティ保護された外部 LDAP Directory Server を使用する場合の Portal Server の設定

デフォルトインストールでは、Sun Java™ System Portal Server、Sun Java™ System Access Manager、および Sun Java™ System Directory Server の各ソフトウェアすべてが同じホストで動作します。ただし、ソフトウェア配備のパフォーマンス、セキュリティ、および統合要件に応じて、Directory Server を別の外部ホストで稼働させ、SSL (Secure Sockets Layer) によるセキュリティ保護された接続を介して Portal Server からディレクトリにアクセスすることもできます。セキュリティ保護された接続を介して Directory Server にアクセスするためには、ディレクトリの証明書に署名した認証局を信頼するように Sun Java™ System Web Server または Sun Java™ System Application Server を設定する必要があります。

Sun Java System Portal Server を設定して、外部 LDAP ディレクトリを使用させるには、次の手順を実行する必要があります。

- Portal Server のインストール。
- SSL を実行するための Directory Server の設定。
- 証明書データベースの作成。
- root 認証局 (CA) 証明書のインストール。
- Directory Server での SSL の有効化。

Directory Server の SSL 設定

1. Directory Server (ns-slapd プロセス) と管理サーバー (ns-httpd プロセス) の両方が起動し、動作していることを確認します。

2. root としてターミナルウィンドウで操作し、次のように入力して Directory Server を起動します。

```
/var/opt/mps/serverroot/startconsole
```

3. 表示されるログインウィンドウで、ユーザー名 admin と、Directory Server のパスワードを入力します。
4. コンソールの左のパネルで、「Server Group」の下に Directory Server インスタンスが表示されるまでディレクトリを展開します。
5. Directory Server インスタンスを選択して「開く」をクリックします。
6. 「タスク」を選択し、「証明書の管理」を選択します。

このタスクを初めて実行するときには、パスワードを入力して証明書データベースを作成するよう求めるメッセージが表示されます。後で Directory Server を起動するときに必要なので、このパスワードをメモしてください。

7. 「要求」をクリックします。

証明書要求ウィザードが表示されます。ウィザードの指示に従い、手順を完了して証明書要求を生成します。証明書要求は、承認を受けるために CMS (証明書管理サーバー) に送信されます。CMS から正式な証明書が返信されます。要求データをファイルにコピーして、証明書要求のコピーを保存します。

8. 証明書要求が CMS に送信され、CMS の管理者により要求が承認されると、承認済みの証明書が返信されてきます。
9. 生成した DS の証明書と CMS 証明書を取り込みます。

CMS は DS 用に証明書を生成したので、root CA として CMS の証明書をインポートすることによって、CMS も信頼される必要があります。

10. 「証明書の管理」、「サーバー証明書」の順に選択して、「インストール」をクリックします。

証明書インストールウィザードが表示されます。

11. [手順 8](#) の承認済みの証明書データをコピーしてテキスト領域に貼り付け、ウィザードの指示に従って証明書をインストールします。

証明書が正常にインストールされると、証明書が行項目として「サーバー証明書」タブに表示されます。

12. 「証明書の管理」ウィンドウを開き、「CA 証明書」タブを選択します。

[手順 9](#) で証明書を発行した CA が CA 証明書リストにある場合は、リストに証明書をインストールする必要はありません。

証明書が証明書リストにない場合は、認証局から root CA 証明書を入手してインストールする必要があります。

- a. 「インストール」をクリックします。
- b. CMS 証明書データをコピーしてテキスト領域に貼り付け、ウィザードの指示に従って証明書をインストールします。

証明書名が CA 証明書リストに表示されます。

13. 「閉じる」をクリックして「証明書の管理」ウィンドウを閉じます。
14. 「設定」タブを選択します。
15. 「暗号化」タブをクリックし、「この暗号化方式ファミリーを使用:RSA」チェックボックスにチェックマークを付け、「保存」をクリックします。
16. 「ネットワーク」タブで、「セキュリティ保護されたポート」フィールドのポート番号が有効かどうかを確認、またはフィールドに有効なポート番号を入力して、「保存」をクリックします。
デフォルトポートは 636 です。
17. Directory Server を再起動し、手順 6 で入力した証明書データベースのパスワードを入力します。

Directory は、SSL 接続を待ち受けて 636 (デフォルト) ポートでの待機を開始します。

信頼データベースを作成する

信頼データベースを作成するときには、鍵ペアファイルで使用するパスワードを指定します。このパスワードは、暗号化通信を使用してサーバーを起動する場合にも必要になります。

証明書データベースで、鍵ペアファイルと呼ばれる公開鍵と秘密鍵を作成して保管します。鍵ペアファイルは SSL 暗号化で使用されます。また、サーバー証明書を要求してインストールする場合にも鍵ペアファイルを使用します。証明書は、インストール後に証明書データベースに保管されます。

証明書データベースの作成手順は、使用している Web コンテナの種類によって異なります。次の説明は、Sun Java System Application Server で証明書データベースを作成する場合の手順です。同じ手順は、Web サイト <http://docs.sun.com> から入手できる『Sun Java System Application Server Administration Guide to Security』でも説明されています。

Sun Java System Web Server で証明書データベースを作成する方法については、Web サイト <http://docs.sun.com> で入手可能な『Sun Java system Web Server, Enterprise Edition Administration Guide』で説明されています。

Sun Java System Application Server で証明書データベースを作成するには、管理インタフェースで次の手順を実行します。

1. Application Server インスタンスが起動していることを確認します。
2. 「アプリケーションサーバーインスタンス」にアクセスし、サーバーインスタンスを選択します。
3. 「セキュリティ」にアクセスします。
4. 「データベースの管理」をクリックします。
5. 「データベースを作成」リンクをクリックします。
「信頼データベースの初期化」ページが表示されます。
6. データベースのパスワードを入力します。
7. もう一度パスワードを入力します。
8. 「了解」をクリックします。
9. 左のパネルで「アプリケーションサーバーインスタンス」およびサーバーインスタンスにアクセスし、「変更を適用」をクリックします。
10. サーバーを停止して再起動し、変更を有効にします。

password.conf ファイルを使用する

SSL が有効に設定されている場合に、SSL/TLS が有効な Sun Java System Application Server をユーザー操作なしで再起動できるようにするには、信頼データベースのパスワードを password.conf ファイルに保存します。

注 システムを適切に保護し、ファイルや鍵データベースが侵害されることのないように注意してください。

password.conf ファイルの詳細については、『Sun Java System Application Server Administrator s Configuration File Reference』の「Using the password.conf File」を参照してください。

通常、サーバーは起動前にパスワードを入力する必要があるため、/etc/rc.local ファイルまたは /etc/inittab ファイルを使用して、UNIX SSL が有効なサーバーを起動することはできません。プレーンテキストにパスワードを保存して自動的に SSL が有効なサーバーを起動することは可能ですが、これは推奨されていません。サーバーの password.conf ファイルは、root またはサーバーをインストールしたユーザーが所有し、所有者だけが読み取りと書き込みのアクセス権を持つようにしてください。UNIX で、SSL が有効なサーバーのパスワードを password.conf ファイルに保存する

ことは、大きなセキュリティリスクとなります。ファイルにアクセス可能なユーザーは、SSL が有効なサーバーのパスワードにもアクセスできます。SSL が有効なサーバーのパスワードを password.conf ファイルに保存する前に、セキュリティ面のリスクを考慮してください。

root 認証局 (CA) 証明書をインストールする

root CA 証明書のインストール手順は、使用している Web コンテナの種類によって異なります。

次の説明は、Sun Java System Application Server に root CA をインストールする場合の手順です。同じ手順は、Web サイト <http://docs.sun.com> で入手できる『Sun Java System Application Server Administration Guide to Security』でも説明されています。

Sun Java System Web Server に root CA をインストールする方法については、Web サイト <http://docs.sun.com> で入手可能な『Sun Java System Web Server, Enterprise Edition Administration Guide』で説明されています。

証明書を発行したソースは、root CA 証明書の入手元となったソースと同じです。

CA からの証明書をインストールするには、管理インタフェースで次の手順を実行します。

1. 「アプリケーションサーバーインスタンス」にアクセスし、左のパネルでサーバーインスタンスを選択します。
2. 「セキュリティ」にアクセスします。
3. 「証明書管理」を選択します。
4. 「インストール」リンクをクリックします。
「サーバー証明書をインストール」が表示されます。
5. クライアント認証の場合に信頼される CA として受け入れる、CA の証明書に対応する「信頼できる証明書発行局」を選択します。
6. ドロップダウンリストから、暗号化モジュールを選択します。
7. 鍵ペアファイルのパスワードを入力します。
8. 次の条件に該当する場合を除いて、それがサーバーインスタンスで使用される唯一の証明書である場合は、証明書名の入力フィールドを空白のまま残します。
 - 複数の証明書が仮想サーバーで使用される。この場合は、サーバーインスタンス内で一意の証明書名を入力します。
 - 内部以外の暗号化モジュールが使用される。この場合は、単一暗号化モジュール内のすべてのサーバーインスタンスの中で一意の証明書名を入力します。

名前を入力すると、「証明書の管理」リストに表示されます。分かりやすい名前を使用してください。たとえば、United States Postal Service CA は CA の名前ですが、VeriSign Class 2 Primary CA は、CA と証明書のタイプの両方を記述しています。

注 証明書名が入力されない場合は、デフォルト値が適用されます。

9. 次のいずれかを選択します。
 - このファイルのメッセージ。この場合は、保存されたメールへのフルパス名を入力します。
 - メッセージテキスト (ヘッダー付き)。この場合は、メールテキストを貼り付けます。テキストをコピーして貼り付ける場合は、先頭と末尾のハイフンなど、Begin Certificate ヘッダーと End Certificate ヘッダーを必ず含めます。
10. 「了解」をクリックします。
11. 「インストール」を選択して新規証明書をインストールします。
12. 左のパネルで「アプリケーションサーバーインスタンス」およびサーバーインスタンスにアクセスし、「変更を適用」をクリックします。
13. サーバーを停止して再起動し、変更を有効にします。証明書は、サーバーの証明書データベースに保管されます。ファイル名は cert8.db です。

Directory Server との通信に Access Manager の SSL 利用を有効にする

Directory Server で SSL を有効化するには、`/etc/opt/SUNWam/config/AMConfig.properties` ファイルを編集します。この手順はコンテナによって異なり、Sun Java System Web Server と Sun Java System Application Server の両方で実行する必要があります。

`AMConfig.properties` ファイルで、次の設定を変更します。

```
com.iplanet.am.directory.ssl.enabled=false
com.iplanet.am.directory.host=server12.example.com (if it needs to be changed)
com.iplanet.am.directory.port=389
```

上記の設定を次のように変更します。

```
com.iplanet.am.directory.ssl.enabled=true
com.iplanet.am.directory.host=server1.example.com
com.iplanet.am.directory.port=636 (port on which DS uses encryption)
```

AccessManager-base/SUNWam/config/ums/serverconfig.xml ファイルの接続ポートと接続タイプの値を変更し、オープンモードから SSL に変更します。

serverconfig.XML ファイルを編集し、次の行を変更します。

```
<Server name="Server1" host="gimli.example.com"
port="389"
type="SIMPLE" />
```

上記の設定を次のように変更します。

```
to
<Server name="Server1" host="gimli.example.com"
port="636"
type="SSL" />
```

serverconfig.xml ファイルに変更を加えた後、Web コンテナを再起動します。

root 以外のユーザーとして Portal Server を実行するための設定

インストール後に実行する次のオプション手順は、root ユーザーとして実行されている Sun Java™ System Portal Server インストールを、root 以外のユーザーとして実行するための設定手順を示しています。

注 Sun Java™ System Enterprise インストーラは、root 以外のユーザーとしての Sun Java™ System Web Server または Sun Java™ System Application Server のインストールをサポートしていません。ただし、root 以外のユーザーとしての Sun Java™ System Directory Server のインストールはサポートしています。この手順では、Web コンテナと Sun Java™ System Directory Server が root 以外のユーザーとして動作していることを想定しています。

特に指定されない限り、すべての手順をスーパーユーザーとして実行してください。Sun Java™ System Portal Server ソフトウェアをインストールした後、次の手順を使用して、Portal Server を root 以外のユーザーとして実行するよう設定します。

1. 次のディレクトリの所有権を、root から *Userid:UserGroup* に変更します。つまり、次のように入力します。
 - `chown -R Userid:UserGroup /opt/SUNWps`
 - `chown -R Userid:UserGroup /etc/opt/SUNWps`
 - `chown -R Userid:UserGroup /var/opt/SUNWps`
 - `chown -R Userid:UserGroup /opt/SUNWam`
 - `chown -R Userid:UserGroup /etc/opt/SUNWam`
 - `chown -R Userid:UserGroup /var/opt/SUNWam`
 - `chown -R Userid:UserGroup WEBCONTAINER-DIR`

Java Enterprise System インストーラーを使用せずに、root 以外のユーザーとして Sun Java System Identity Server をインストールした場合、Access Manager のディレクトリの変更方法については、Access Manager のマニュアルを参照してください。

2. Portal Server のディレクトリに次の権限を設定します。

- `chmod 0755 /opt/SUNWps`
- `chmod 0755 /etc/opt/SUNWps`
- `chmod 0755 /var/opt/SUNWps`
- `chmod 0755 /opt/SUNWam`
- `chmod 0755 /etc/opt/SUNWam`
- `chmod 0755 /var/opt/SUNWam`
- `chmod 0755 WEBCONTAINER-DIR`

3. Directory Server を root 以外のユーザーとして再起動します。

Java Enterprise System インストーラーによって、`/usr/jdk/entSYS` に Java™ Development Kit (JDK™) がインストールされます。ディレクトリの所有権と権限を、次のように root 以外のユーザーに変更します。

- `chown -R Userid:UserGroup /usr/jdk`
- `chmod 0755 /usr/jdk`

4. Web コンテナと Directory Server を停止します。

5. すべてのプロセスが停止していることを確認してください。

確認するには、次のように入力します。

- `ps -aef | grep slapd`
- `ps -aef | grep httpd`
- `ps -aef | grep http`
- `ps -aef | grep admin`

6. シャットダウンされなかったプロセスは強制終了してください。

7. Directory Server と Web コンテナを起動します。

8. ディレクトリと Web コンテナプロセスの所有者を確認します。root 以外のユーザーとして動作しています。

注 root 以外のユーザーとして Portal Server を実行しているときにパッチを適用する場合は、まず Portal Server ディレクトリの所有権を root 以外からスーパーユーザー (root) に戻す必要があります。パッチが正常に適用された後、root 以外のユーザーとして Portal Server を実行するための所有権と権限をもう一度設定してください。

Sun Java System Portal Server の管理

第 6 章 「認証、ユーザー、およびサービスの管理」

第 7 章 「管理の委任の設定」

第 8 章 「ポータルデスクトップサービスの管理」

第 9 章 「Web Services for Remote Portlets (WSRP) サービスの管理」

第 10 章 「ディスプレイプロファイルの管理」

第 11 章 「NetMail サービスの管理」

第 12 章 「リライトサービスの管理」

第 13 章 「検索エンジンサービスの管理」

第 14 章 「検索エンジンロボットの管理」

第 15 章 「定義済みのロボットアプリケーション関数」

第 16 章 「登録サービスの管理」

第 17 章 「通信チャネルの設定」

認証、ユーザー、およびサービスの管理

Sun Java™ System Access Manager を使用して、認証、ユーザー、およびサービスを管理する方法を説明します。この章では、Access Manager のすべての特徴については説明していません。Sun Java™ System Portal Server に関する側面に重点を置いていません。詳細については、Access Manager のマニュアルを参照してください。

この章で説明する内容は次のとおりです。

- [Sun Java System Access Manager の概要](#)
- [Access Manager 管理コンソールへのログイン](#)
- [基本情報の表示](#)
- [Portal Server の起動と停止](#)
- [Access Manager サービスの管理](#)
- [Portal Server ユーザーの管理](#)
- [認証の設定](#)
- [Portal Server によるポリシー管理の使用法の概要](#)
- [Portal Server デスクトップへのログイン](#)
- [ロギングの管理](#)

Sun Java System Access Manager の概要

Sun Java System Portal Server (旧 Sun™ ONE Portal Server) の実装では、認証方法の管理、ドメイン、ロール、およびユーザーの作成、プロファイルの属性やログなどのその他のデータの管理に、この製品を使用します。また、カスタムアプリケーションの開発には iPlanet Portal Server 3.0 API も使用します。

Portal Server 6 製品では、以前は iPlanet Portal Server 3.0 によって提供されていた Access Manager の管理機能と API を使用します。Access Manager は、Sun Java™ System Directory Server の管理機能とセキュリティ機能を活用するツールセットです。Access Manager の目的は、Sun Java System Directory Server を使用する組織に、ユーザーオブジェクト、ポリシー、およびサービスを管理するためのインタフェースを提供することです。

Access Manager を使用することで、次の操作を行えます。

- Sun Java System Directory Server による、データセキュリティを向上させたユーザー認証およびシングルサインオンの実行。
- 管理者による、ロール (ユーザーエントリの属性として表現される、エントリのグループ化の仕組み) に基づくユーザーエントリ管理。
- 開発者による、多数のデフォルトサービスおよびカスタムサービスの設定パラメータの定義と管理。

この 3 つの機能には、Web ベースのグラフィカルユーザーインタフェースである Access Manager 管理コンソールからアクセスします。また、コマンドラインインタフェース (amadmin) を利用して、ディレクトリサーバーでバッチ管理タスクを実行できます。たとえば、新規サービスの作成、追加、有効化や、組織、ユーザーコンテナ、グループ、ロール、ユーザーの作成、削除、および読み取り (取得) を実行できます。

Access Manager の機能の概要

Access Manager には、次の管理コンポーネントがあります。従来は、これらのコンポーネントは Portal Server 3.0 のフレームワーク自体に含まれていました。

- **ユーザー管理**: ユーザー関連オブジェクト (ユーザー、ロール、グループ、ピープルコンテナ、組織、サブ組織、部署のオブジェクト) を作成、管理します。これらのオブジェクトは、Access Manager コンソールまたはコマンド行インタフェースのいずれかを使用して、定義、変更、削除できます。
- **認証**: ユーザー認証のプラグインソリューションを提供します。特定のユーザーを認証するのに必要な条件は、Portal Server エンタープライズの各組織に設定された認証サービスに基づきます。Portal Server セッションへのアクセスが許可されるには、ユーザーは認証を受け、それを通過する必要があります。

- **シングルサインオン**: ユーザーが認証されると、Access Manager のシングルサインオン (SSO) 用 API が、認証を継承します。認証されたユーザーが保護されたページへのアクセスを試みると、そのたびに SSO API は認証証明情報に基づいて、ユーザーが適切なアクセス権を持っているかどうかを判断します。ユーザーが有効であれば、追加認証なしでページへのアクセスが許可されます。無効であれば、ユーザーに再認証が求められます。
- **サービス管理**: Portal Server 製品自体のサービス (ポータルデスクトップ、リライタ、検索、および NetMail) を含む、デフォルトサービスとカスタムサービスの設定パラメータを指定します。
- **ポリシー管理**: ビジネスリソースへのアクセスを制御するルールを定義、変更、または削除します。ポリシーとは、これらのルールの総称です。ポリシーは、ロールまたは組織に基づき、アクセス権を与えたり、制約を定義したりすることができます。

Portal Server 3.0 と Portal Server 6.2 の比較

91 ページの表 6-1 は、Portal Server 製品の主な変更の概要を示しています。Access Manager には、Sun ONE Portal Server 3.0 (旧 iPlanet Portal Server 3.0) 製品の一部であった多くの特徴と機能が含まれています。この表では、1 列目に概念または用語を示し、2 列目に Portal Server 3.0 製品でのその用語の特徴や機能を定義し、3 列目で Portal Server 6.2 製品での対応する特徴や機能について説明します。

注 これらの変更は Sun Java System 2003Q4 製品出荷と同時に行われ、この情報はその製品のユーザー用に保持されています。

表 6-1 Portal Server 3.0 と Portal Server 6.2 の比較

概念または用語	Portal Server 3.0	Portal Server 6.2
ロールツリー	<p>ユーザーおよびアプリケーションを組織化するために Portal Server 3.0 内で設定する階層。ロールツリーには、次の 4 つのレベルがあります。</p> <ul style="list-style-type: none"> • root • ドメイン • ロール • ユーザー 	<p>ロールツリーの概念は適用されなくなります。</p> <p>代わりに、Access Manager は、Sun Java System Directory Server の機能を利用するため、ユーザー、組織、サブ組織など組織化にはディレクトリ情報ツリー (DIT) を使用します。</p>

表 6-1 Portal Server 3.0 と Portal Server 6.2 の比較 (続き)

概念または用語	Portal Server 3.0	Portal Server 6.2
ドメイン / 組織	社員または顧客など、一般的な関係を持つユーザーの最上位グループ。これは DNS ドメインではありませんが、Portal Server 3.0 が、ユーザーを論理コミュニティにグループ化するために使用します。	ドメインの概念は適用されなくなります。代わりに、部門とリソースを管理するために企業が使用する階層構造の最上位レベルは、Access Manager では組織となります。 インストール時に、Access Manager はルートサフィックスを要求し、デフォルトはドメイン名から作成されます(たとえば、ドメインが sun.com であれば、デフォルトは dc=sun, dc=com)。インストール後に、追加の組織を作成して、別の企業を管理することができます。作成された組織はすべて最上位組織の下に配置されます。また、これらのサブ組織内には、別のサブ組織を入れ子にすることができます。入れ子の構造の深さに制限はありません。
ロール	機能に基づいてドメインのメンバーを分けます。ロールには、ユーザーの Desktop ポリシーを定義する属性セットとポリシーが含まれます。	1 人または複数のユーザーに付与できる、単一の権限や権限のセットが含まれます。これには、Sun Java System Directory Server に保存された ID 情報のアクセスと管理、および Access Manager ポリシーモジュールによって保護された権限へのアクセスが含まれます。また、Access Manager のロールは、ロール自体をサービスクラスのテンプレートに保存されているプロファイルと関連付けます。 Access Manager でのロールの定義方法は異なり、従来はサポートされていなかった、単一ユーザーへの複数のロールの割り当ても行うことが可能です。 アクセス権はアクセス制御命令 (ACI) によってロールに定義されます。Access Manager には、いくつかのロールが事前に定義されています。Access Manager コンソールを使用してロールの ACI を編集し、ディレクトリ情報ツリー内でアクセス権を割り当てることができます。

表 6-1 Portal Server 3.0 と Portal Server 6.2 の比較 (続き)

概念または用語	Portal Server 3.0	Portal Server 6.2
属性	<p>グローバル属性とユーザー設定属性の 2 種類がサポートされます。グローバル属性はプラットフォーム全体に適用され、スーパー管理者だけが設定できます。ユーザー設定属性は、後述するように、ロールツリーの基本レベルに適用されます。代理のドメイン管理者は、ドメイン、親ロール、子ロール、およびユーザーレベルにこれらの属性を設定できます。属性によっては、ロールツリーのユーザーレベルで必要に応じてユーザーごとにカスタマイズできます。</p>	<p>Access Manager の属性には、次のような種類があります。</p> <ul style="list-style-type: none"> • グローバル: グローバル属性に適用される値は、Access Manager の設定全体に適用され、設定済みのすべての組織に継承されます。 • ダイナミック: ダイナミック属性は、Access Manager の設定済みロールまたは組織に割り当て可能です。ユーザーにロールを割り当てる場合、または組織でユーザーを作成する場合は、ダイナミック属性がそのユーザーの特性となります。 • 組織: この属性は、組織だけに割り当てられます。その意味では、この属性はダイナミック属性として機能します。ただし、ダイナミック属性とは異なり、これはサブツリーのエントリーによって継承されます。 • ユーザー: この属性は各ユーザーに直接割り当てられます。この属性は、ロールまたは組織から継承されず、通常はユーザーごとに異なります。 • ポリシー: ポリシー属性は権限属性です。ポリシーを設定すると、ロールまたは組織に、それを割り当てることができます。これは、ダイナミック属性とポリシー属性の唯一の違いです。ダイナミック属性は、ロールまたは組織に直接割り当てられますが、ポリシー属性はポリシーの設定に使用され、その上でロールまたは組織に割り当てられます。

表 6-1 Portal Server 3.0 と Portal Server 6.2 の比較 (続き)

概念または用語	Portal Server 3.0	Portal Server 6.2
Policy	アプリケーション、Desktop、NetFile、Netlet などに対するポータルアクセスポリシーを設定します。	<p>どのリソースに誰が何を行うかを定義するルールです。Access Manager のポリシーサービスにより、組織はこれらのルールやポリシーを設定できます。通常、ポリシーは組織 (またはサブ組織) レベルで作成され、組織のツリー全体で使用されます。指定のポリシーを作成するには、ポリシーの作成先となる組織に特定のポリシーサービスが最初に追加されている必要があります。</p> <p>Sun Java System Identity Server 6.2 では、ポリシーサービスには、許可または拒否される URL のリストだけが含まれます。これでは、Portal Server がコンテンツ用にポリシーベースの Desktop を構築するには不十分です。このため、チャンネルアクセスのポリシーが、Desktop のディスプレイプロファイルに組み込まれます。Portal Server 6 の Desktop は、複数のロールからチャンネルリストをマージできるディスプレイプロファイルをサポートしています。たとえば、25 のロールがあり、それぞれにそのロールに関連する少数のチャンネルがある場合、任意の数のロールを持つようにユーザーを設定し、次に、ユーザーが取得する Desktop がそのロールセットを提供するように設定できます。各種ロールからのチャンネルの集約やマージは、マージ動作によって制御されます。ディスプレイプロファイルをマージするために、Portal Server のロールには階層的な順序が必要になります。マージは、優先度が最も低いドキュメント (もっとも小さい番号) から開始され、優先度が最も高いプロファイルであるユーザー (ベース) に達するまで、優先度番号の昇順に行われます。ディスプレイプロファイルのマージについては、第 10 章「ディスプレイプロファイルの管理」を参照してください。</p>

表 6-1 Portal Server 3.0 と Portal Server 6.2 の比較 (続き)

概念または用語	Portal Server 3.0	Portal Server 6.2
コンポーネント / サービス	Portal Server 3.0 の 4 つの主要コンポーネントは、サーバー自体、プロファイラサーバー、ゲートウェイ、ファイアウォールです。	コンポーネントは、共通の名前で定義された属性のグループである Access Manager サービスに置き換えられました。属性は、サービスが組織に提供するパラメータを定義します。 Access Manager はサービスのフレームワークです。 Portal Server 6 は、認証、ユーザー管理、およびポリシー管理などのコアサービスの提供だけでなく、Portal Server 固有のサービス (Desktop、NetMail、リライタ、および検索) を実行するフレームワークについても Access Manager に依存します。
管理インタフェース	Portal Server 3.0 コンポーネントだけを管理するための、専用の管理コンソールを提供します。 コマンド行インタフェースは、 <code>ipsadmin</code> です。	Access Manager のサービス、ユーザー、およびポリシーの管理だけでなく、Portal Server 固有のサービス (Desktop、NetMail、リライタ、および検索) の管理にも Access Manager 管理コンソールを使用します。 <code>ipsadmin</code> に代わるコマンド行インタフェースは、 <code>amadmin</code> 、 <code>dpadmin</code> 、および <code>rwadmin</code> です。

Portal Server 6.0 と Portal Server 6.2 の比較

96 ページの表 6-2 は、Portal Server 6.0 から Portal Server 6.2 への改訂で加えられた変更の概要を示しています。この表では、1 列目に概念または用語を示し、2 列目に Portal Server 6.0 製品でのその用語の特徴や機能を定義し、3 列目で Portal Server 6.2 製品での対応する特徴や機能について説明します。

注 これらの変更は Sun Java System 2003Q4 製品出荷と同時に行われ、この情報はその製品のユーザー用に保持されています。

表 6-2 Portal Server 6.0 と Portal Server 6 の比較

概念または用語	Sun Java System Portal Server 6.0	Portal Server 6
ポリシー	ユーザーにポリシーを割り当てます。一度ポリシーを指定し、作成すると、組織またはロールを割り当てることができます。組織レベルでポリシーを割り当てると、組織のすべてのエントリーで属性が有効になります。ロールにポリシーを割り当てると、ロールの属性を持つすべてのユーザーで属性が有効になります。	組織のポリシーに関する定義と決定を、別の組織に委託します。または、リソースのポリシー定義を別のポリシー製品に委託することもできます。この、ポリシーの作成と評価に関するポリシー委任は、参照ポリシーによって制御されます。 アクセス権の定義には標準のポリシーを使用します。標準のポリシーは、複数のルール、サブジェクト、条件から構成されます。
認証メニュー	Sun ONE Identity Server 5.1 管理コンソールの認証メニューの設定機能は、ユーザーが選択した認証モジュールのメニューをサポートします。	有効な認証モジュールを選択できるリストを設定する場合は、Sun Java System Identity Server の管理コンソールを使用して、認証レベル属性で各認証モジュールが同じ値を持つように設定します。認証モジュールの設定については、 第 6 章「認証、ユーザー、およびサービスの管理」 を参照してください。

Access Manager の制約

Access Manager を使用する場合は、次の制約があります。

- 事前定義された Access Manager ロールは、複数の並列組織に割り当てることができません。ただし、ロールは、ロールが関連する組織の子組織に属するユーザーに割り当てることができます。また、複数ドメインのリソースへのアクセスは、カスタムロールを作成し、また必要な ACI を定義して、必要な権限をロールに付与することにより可能になります。
- ユーザーは、組織に属する必要があるため、その組織にだけ属することができます。
- 階層型ロールはサポートされません。たとえば、ロール A およびロール B の合計に等しいものとしてロール C を作成することはできず、また、ロール C を持つユーザーは、明示的にロール A に割り当てられていない限り、ロール A のリソースにアクセスできません。
- RoleAdministratorRole のアクセス権は、対応する ACI を直接編集した場合のみ設定できます。

- ロール管理者 (代理の管理者) が Access Manager 管理コンソールにログインする場合、ロール管理者に修正する権限がない場合でも、同じ組織にあるロール、および関連するサービスとプロパティのすべてを表示できます。

Access Manager インタフェース

Access Manager 管理コンソール

このブラウザベースのコンソールは、グラフィカルユーザーインターフェースを提供して、Portal Server サービスなどの Access Manager エンタープライズを管理します。管理コンソールには、サービス、ポリシー、およびユーザーの作成および管理に使用するさまざまなレベルの権限を持つデフォルトの管理者が含まれます (代理の管理者を追加する場合は、ロールに基づいて管理者を作成できる)。詳細については、[第7章「管理の委任の設定」](#)を参照してください。

Access Manager 管理コンソールは、3つのセクションに分割されています。ロケーションパネル、ナビゲーションパネル、およびデータパネルです。3つのパネルを使用して、ディレクトリを移動したり、ユーザーおよびサービスの設定を実行したり、またポリシーを作成したりします。

詳細については、[第1章「Sun Java System Portal Server の管理の概要」](#)を参照してください。

Access Manager コマンド行

Access Manager コマンド行インターフェースは、サーバーを管理する amadmin です。また、amadmin は、XML サービスファイルをディレクトリサーバーにロードし、ディレクトリツリーでバッチ管理タスクを実行するのに使用します。iPlanet™ Portal Server 3.0 のコマンド行インターフェース、ipsadmin および ipsserver は、現在、使用されていません。

amadmin については、Access Manager のマニュアルを参照してください。

Access Manager 管理コンソールへのログイン

Access Manager コンソールへは、次のどちらかの方法でログインできます。

- 固有の URL を使用する
- HTTPS を経由する

管理コンソールにログインする場合、表示される機能は、ユーザーのアクセス権によって異なります。アクセス権は、ACI あるいはユーザーに割り当てられたロールによって決まります。たとえば、スーパーユーザーには、管理コンソールのすべての機能が表示されます。一方、代理管理者には、通常、サブ組織についてこの機能のサブセットだけが、またエンドユーザーには、自分のユーザー ID に関するユーザー属性だけが表示されます。

現在、管理コンソールへは、次のどちらかの URL でログインできます。

- `http://host:port/amconsole/`
- `http://host:port/amserver/`

URL `/amconsole` は、Access Manager 管理コンソールの HTML ページを明示的に要求します。`/amconsole` を使用してログインすると、管理コンソールが起動して、URL が `/amserver/UI/login` に変更されたことが確認でき、ユーザーは認証を行えます。設定に関係なく、この URL を管理コンソールへのアクセスに使用することができます。

URL `/amserver` は、Access Manager サービスの HTML ページを要求します。Portal Server のインストールの際にセットアップされるデフォルトでは、この URL にリダイレクトして管理コンソールにログインすることになっていますが、URL `/amserver` は、Access Manager サービスにアクセスするため、この URL はコンソール以外のその他のサービスを利用可能にするために使用できます。たとえば次の場合に使用します。

- ユーザーが無効なセッションを持つアプリケーションにアクセスする場合、アプリケーションは、URL `/amserver` 要求を `goto` パラメータで `amserver/UI/login` に転送する場合があります。たとえば、Portal Server Desktop は、Access Manager エージェントと同様にこの転送を行います。
- 顧客は、アプリケーションまたはポータルへの開始位置として `amserver/UI/login` にユーザーを転送する場合があります。また、デフォルトのリダイレクト URL が、ポータルアプリケーションまたはカスタムアプリケーションになる場合があります。
- カスタムアプリケーションが、`amserver/UI/login` を直接呼び出し、認証する場合があります。

Access Manager 管理コンソールにログインするには次のようにします。

- 固有の URL を使用する : `http://host:port/amserver/` を入力します。
または `http://host:port/amconsole/` を入力します。
- HTTPS を使用する : `https://host:ssl_port/amconsole/` を入力します。

IP アドレスを使用した管理コンソールへのログインの設定

サーバーの IP アドレスを使用して Access Manager 管理コンソールにログインすることはできません。これは Access Manager での cookie のドメイン設定によるものです。

ただし、管理コンソールの「Cookie ドメイン」リストに、ローカルホストの IP アドレスを追加することはできます。

1. ロケーションパネルで「サービス設定」を選択します。
2. 「プラットフォーム」をクリックします。
3. 「グローバル」にローカルホストの IP アドレスを追加します。

これで、ドメイン名ではなく IP アドレスで管理コンソールにアクセスできます。

基本情報の表示

スクリプトにより、`jar` ファイルのバージョンおよびビルドの日付だけでなく、`Portal Server` のバージョンやビルドの日付など、製品についての基本情報を表示することができます。バージョンスクリプトは、`PortalServer-base/SUNWps/bin` ディレクトリにインストールされます。ここで `PortalServer-base` は、`Portal Server` をインストールしたベースディレクトリです。デフォルトは `/opt` です。

製品情報を表示するには、次の手順に従います。

1. スクリプトがインストールされているディレクトリに移動します。次のように指定します。

```
cd PortalServer-base/SUNWps/bin
```

2. `Portal Server` についての情報を表示するには、次のように入力します。

```
./version
```

3. `Portal Server` の `jar` ファイルについての情報を表示するには、次のように入力します。

```
./version jar-file
```

ここで `jar-file` は `jar` ファイルの名前です。

Portal Server の起動と停止

ここでは、Portal Server の停止および起動方法について説明します。それぞれの Web コンテナ用のスクリプトを使用して、その Web コンテナインスタンスを再起動する必要があります。たとえば、次のようになります。

- Sun Java System Web インスタンスを起動するには、次のスクリプトを使用します。

```
ws-install-base/https-instancename/start
```

- Sun Java System Web インスタンスを停止するには、次のスクリプトを使用します。

```
ws-install-base/https-instancename/stop
```

- Sun Java System Application Server を起動するには、次のスクリプトを使用します。

```
cd /var/opt/SUNWappserver7/domains/domain1
```

```
./asadmin asadmin> start-domain --user admin domain1 asadmin>  
exit
```

注 再起動する前に、サーバーを停止する必要はありません。すでに起動中のサーバーを起動すると、そのサーバーは停止してから再起動します。

これらの操作は、Web コンテナによって異なります。詳細については、Web コンテナのマニュアルを参照してください。

Portal Server は、さまざまなプラットフォームのロケールをサポートしています。Portal Server をインストール済みのデフォルト値以外で起動する方法については、『Sun Java System Portal Server 6 2005Q1 Developer's Guide.』を参照してください。

Access Manager サービスの管理

ここでは、Portal Server が使用する Access Manager サービスについて説明します。詳細については、Access Manager のマニュアルを参照してください。

インストールおよび Sun Java System Web Server パッケージ

- Portal Server インストーラは、Access Manager がインストールされていない場合、Access Manager インストーラを実行します。
- Portal Server は、Access Manager と Web コンテナを共有します。Web コンテナは、並行性、配備、ライフサイクル管理、セキュリティ、トランザクション、その他のサービスを含む Web コンポーネントの実行時環境を指定します。
- Portal Server は、JVM™ および Access Manager が提供するその他のコンポーネントを使用します。

ユーザー管理

- Portal Server は、Access Manager API を使用して、自身のプロフィール情報を Access Manager に格納します。
- Portal Server は、Access Manager の複数ロールサポートを活用します。
- Portal Server は、たとえば、givenName など、オープンで所有者のいない標準のスキーマの属性を使用します。
- Access Manager は、LDAP ディレクトリへの直接アクセスを提供します。

シングルサインオン / 認証

- Portal Server 6 で、認証は、Access Manager によって管理されます。
- Access Manager は、すべての認証モジュールを提供します。
- Portal Server は、Access Manager のポリシー属性を使用して、アクセスを制限します。

サービス管理

Portal Server 6 は、次の Access Manager サービスを定義します。

- **Desktop:** ポータルフロントエンドを提供する、ポータルへのプライマリエンドユーザーインターフェース。ポータルデスクトップのセットアップおよび管理については、[第 8 章「ポータルデスクトップサービスの管理」](#)を参照してください。
- **NetMail:** インターネット上で IMAP および SMTP メールサーバーにアクセスし、ポータルを通じてユーザーが電子メールにアクセスできるようにします。NetMail のセットアップおよび管理については、[第 11 章「NetMail サービスの管理」](#)を参照してください。
- **リライタ:** 管理者によってセットアップされるルールを実装し、適切にアクセスできるように URL をリライトします。リライタのセットアップおよび管理については、[第 12 章「リライタサービスの管理」](#)を参照してください。
- **検索:** 使用可能なドキュメントの基本および詳細検索チャネルなどの検索機能を Portal Server に提供します。検索サービスのセットアップおよび管理については、[第 13 章「検索エンジンサービスの管理」](#)を参照してください。

Portal Server ユーザーの管理

ディレクトリ情報ツリー (DIT) は、ユーザー、組織、サブ組織などを論理構造または階層構造に編成することにより、これらのロールを想定しているユーザー、または組織内に属しているユーザーへの適切なアクセスを効率的に管理し、割り当てることを可能にします。この節では、組織、サブ組織、ロールの機能と性能についての情報を説明し、また、組織、ロール、ユーザーの作成と管理の手順を示して、Portal Server の実装の基本となるディレクトリ構造またはツリーの設計に役立つ情報について説明します。

注 Portal Server 6 は組織をサポートします。この組織については、以前の Portal Server 3.0 では、ドメインの概念を使用していました。

Access Manager の組織ツリーの最上位は、インストール時に指定されます。インストール後に、追加の組織を作成して、別の企業を管理することができます。作成された組織はすべて最上位組織の下に配置されます。これらのサブ組織内で、他のサブ組織を入れ子にできます。入れ子の構造の深さに制限はありません。

注 ツリーの最上位を `isp` と呼ぶ必要はありません。任意の名前をつけることができます。ただし、たとえば、`isp` など一般的な最上位で編成されたツリーでは、ツリー内の組織はロールを共有することができます。

ロールは、より効果的に、またより簡単にアプリケーションを使用するように設計された新しいグループ化の仕組みです。それぞれのロールはメンバー、あるいはロールを保有するエントリを持ちます。グループの場合と同じく、ロールのメンバーは明示的またはダイナミックに指定できます。ロールの仕組みにより、そのエントリがメンバーになっているすべてのロール定義の DN を含む nsRole 属性が自動的に生成されます。各ロールは、1 人または複数のユーザーに付与できる、単一の権限や権限のセットを含んでいます。Portal Server 6 では、複数のロールを 1 人のユーザーに割り当てることができます。ロールの権限はアクセス制御命令 (ACI) で定義されます。Portal Server には、いくつかのロールが事前に定義されています。Access Manager コンソールを使用してロールの ACI を編集し、ディレクトリ情報ツリー内でアクセス権を割り当てることができます。用意されている例には、Top-level Admin Role および Top-level Help Desk Admin Role が含まれます。組織間で共有できるその他のロールを作成することもできます。

組織、サブ組織、およびロールの設計

DIT の構造を設計する場合、階層を持つツリー構造にするかフラットなツリー構造にするかを定めることが必要です。一般的には、作成するツリーをできる限りフラットにするようにします。ただし、ユーザーアクセスの付与および管理を容易にするには、組織の成長の大きさに合わせて、ある程度の数の階層があることが重要になります。DIT 構造を構築するために必要な Access Manager の主要な 3 つのキー構造エンティティは、組織 (サブ組織)、ロール、およびユーザーです。構造を設計する前に、これらの各エンティティの機能、特性、および相互関係について理解する必要があります。

組織およびサブ組織

- 所属する企業や組織の階層に対し、それを表現し形取った階層関係の作成を可能にします。
- 対応する管理者によって作成される特定のユーザーを含めることができます。これにより、管理およびアクセス制御の目的で、ユーザーをグループ化する方法が提供されます。通常、同じような必要性を持つユーザーをグループ化すると、アクセスの管理および制御がより簡単になります。
- 管理者が管理コンソールを使用して、親組織またはサブ組織を簡単に作成したり、削除したりすることができます。ただし、すべて削除されると、すべてのサブ組織およびユーザーも削除されるため、名前や構造を変更する予定がある場合には適していません。

ロール

- 1人または複数のユーザーへ、単一の権限や権限のセットを割り当てることを可能にします。組織内では、複数ロールを定義して特定の権限をユーザーに設定することができます。
- 直接編集する必要があるアクセス制御命令 (ACI) を通じて許可を定義します。一度定義すると、組織、サブ組織、またはユーザーに、簡単に割り当て、また割り当てを取り消すことができます。1つのエンティティからのロールの割り当ての取り消しは、そのエンティティにだけ適用されます。ロールは引き続き存在し、割り当てられたままで他のエンティティに割り当てし直すことが可能なため、アクセスの変更が頻繁に必要な組織により適しています。
- チャンネルの可視性およびユーザーのチャンネル上書き機能を制御します。XML ディスプレイプロファイル内で、XML ドキュメントのチャンネルの可視または不可視をデフォルトに設定できます。また、XML ドキュメントのデフォルトのチャンネルが上書きされないようにすることもできます。

ユーザー

- 人物の ID を表します。管理者が、組織またはサブ組織内に作成できます。
- 複数のロールに関連付けることができますが、ロールの範囲内にある必要があります。またユーザーは、サブ組織から属性を継承します。
- 1つの組織またはサブ組織にのみ属しますが、管理者に権限があれば、ユーザーのある組織から別の組織に簡単に移動できます。
- チャンネルの可視性をパーソナライズできます。

シナリオ 1: サブ組織とロールから構成される階層構造

構造をできるだけフラットにする必要はありますが、階層によっては、必要なグループ化を行うのに役立ちます。階層構造を作成するための手順の概要は次のとおりです。

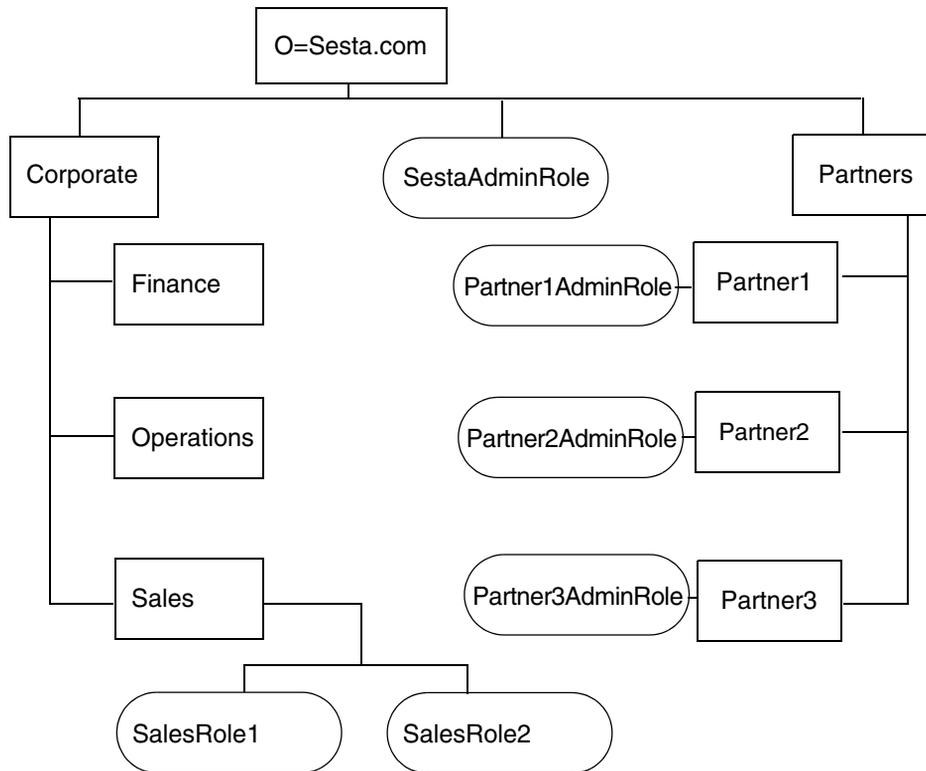
1. 最上位組織を作成します。
2. 企業のユーザーの機能グループまたは組織グループをすべて識別し、DIT 構造エンティティを作成するのに必要なグループ、すなわち固有の権限を持つ必要があるグループを決定します。通常、これは企業で唯一最大の下位部門で、管理者が管理する必要があります。一般的または機能的な名前を使用して、再編成および名前の変更に問題が発生しないようにします。
3. 最上位組織と何らかの関係がある各 DIT エンティティについて、そのエンティティにサブ組織 (つまり、Access Manager 内に存在する別の組織のサブ組織)、またはロールを作成します。

次のガイドラインを使用して、サブ組織またはロールを使用するかどうかを判断します。

- 同様のアクセスの必要性を持つユーザーのグループ化を含むエンティティにサブ組織を定義します。通常、これは、単一組の許可を割り当てることのできる広範囲な機能エンティティ、または組織エンティティです。
 - 子組織のユーザーがこのロールを持つ必要があり得る場合は、ロールを定義します。すべてのユーザーは、1つの組織またはサブ組織に属します。割り当てられたロールがない場合は、常駐する組織から許可を継承します。このため、属する組織および任意の親組織の両方の属性をユーザーが保有するようにする場合は、ロールの仕組みを使用して、複数のロールを割り当てる必要があります。
4. ロールごとに、**RoleAdministratorRole** を定義してロールを管理します。次に **ACI** を適切に設定します (管理権限: ユーザーの追加および削除、ロール属性の修正など)。
 5. 企業にアクセスするユーザーを定義します。ユーザーが組織の権限を継承している場合は、ユーザーを適切な組織に配置します。ユーザーがロールの割り当てを通じて権限を受け取る場合は、ロールの範囲内、つまり組織内に入るか、ロールが定義されている組織の子になるように配置する必要があります。

図 6-1 は、階層型ディレクトリ構造を示しています。この図では、最上位組織は **Sesta.com** です。最上位の1つ下には、組織と **Corporate** と **Partners** のサブ組織を管理する **SestaAdminRole** が置かれます。Corporate 組織には、**Finance**、**Operations**、および **Sales** の3つのサブ組織があります。Sales 組織には、複数のタイプのユーザーが属するため、**SalesRole1** および **SalesRole2** の2つのロールが定義されています。Partners 組織には、**Partner1**、**Partner2**、および **Partner3** の3つのサブ組織があります。これらの組織にはそれぞれ独自の管理者が必要であるため、3つのロールが定義され、それぞれが適切な組織に関連付けられています。パートナーロールは、**PartnerAdmin1**、**PartnerAdmin2**、および **PartnerAdmin3** です。

図 6-1 階層型ディレクトリ構造

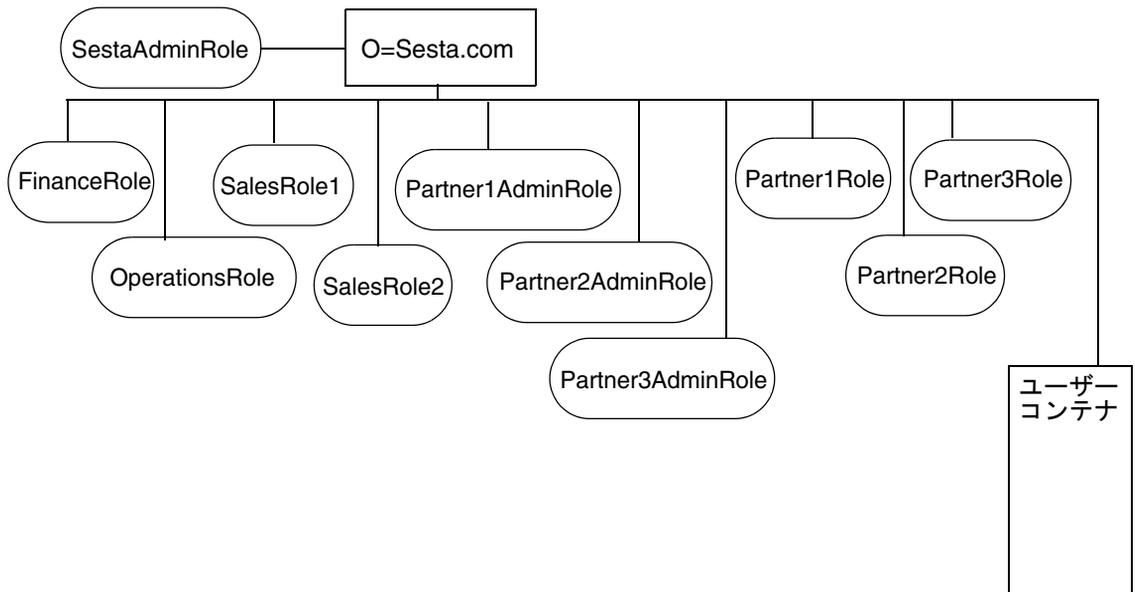


シナリオ 2: フラットなツリー構造

組織を頻繁に変更する場合は、よりフラットな、あるいは全体として一様にフラットなツリー構造が適しています。1つのピープルコンテナを含む1つの組織と、すべてが同じレベルのロールとから構成される構造は、企業を頻繁に変更する際に役立つことがよくあります。1つの組織では、企業の変更が DIT に影響を及ぼすことはありません。すべてのアクセス権は、ロールを使用して定義されます。すべてのユーザーが単一のピープルコンテナに格納され、またすべてのロールが同じレベルにあるため、任意のユーザーに任意のロールを割り当てることができます。

図 6-2 は、フラットなディレクトリ構造を示しています。この図で、最上位かつ唯一の組織は Sesta.com です。すべてのエンティティは、この最上位組織の下に直接定義されます。これらのエンティティには、SestaAdminRole があり、組織や、Finance、Operations、Sales1、および Sales2 ユーザーが必要とするさまざまな企業機能の 4 つのロール、およびパートナーが必要とするユーザー機能の 6 つのロール、Partner1Role、Partner2Role、Partner3Role、Partner1AdminRole、Partner2AdminRole、および Partner3AdminRole を管理します。

図 6-2 フラットなディレクトリ構造



組織およびサブ組織の新規作成

組織とサブ組織を使用すると、管理およびアクセス制御を目的としてユーザーを構造化およびグループ化できます。企業の階層または構造を決定した後、それを実装するために必要な組織とサブ組織を作成する必要があります。組織またはサブ組織を新しく作成する場合、デフォルトで定義されるサービス、ポリシー、ユーザー、およびロールはありません。このため、新しい組織またはサブ組織を作成する場合は、常に、次の手順で設定を行う必要があります。

1. 組織で有効にするサービスをすべて追加します。詳細については、「[サービスを追加する](#)」を参照してください。通常、少なくとも次のサービスを追加します。
 - 認証。コア認証サービスおよび組織のユーザーが認証に使用する任意の認証サービス (LDAP、匿名)。詳細については、「[認証の設定](#)」を参照してください。
 - URL ポリシーエージェント。
 - ユーザー。
 - Portal Server の設定。組織のユーザーに対して有効にする Portal Server の任意のサービス (デスクトップおよび NetMail)。
2. 追加された各サービスにテンプレートを作成します。詳細については、「[サービスのテンプレートを作成する](#)」を参照してください。
3. 組織内のユーザーにアクセス権限を付与する必要があるポリシーを作成します。ポリシーの使用についての詳細は、「[Portal Server によるポリシー管理の使用法の概要](#)」を参照してください。
4. ユーザーを組織に追加します。詳細については、「[新規ユーザーを追加する](#)」を参照してください。
5. 組織に必要な任意のロールを作成し、割り当てます。詳細については、「[新規ロールを作成する](#)」および「[ロールをユーザーに割り当てる](#)」を参照してください。
6. 組織で有効なサービスを設定します。Desktop の設定については、[第 8 章「ポータルデスクトップサービスの管理」](#)を参照してください。NetMail の設定については、[第 11 章「NetMail サービスの管理」](#)を参照してください。

組織を新規作成し、ポータルを使用するように組織を設定する迅速な方法については、「[ポータル組織の迅速な新規作成](#)」を参照してください。

新規の組織またはサブ組織を作成する

Portal Server 用に、組織およびサブ組織を設計する推奨の方法については、「[組織、サブ組織、およびロールの設計](#)」を参照してください。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。

デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. サブ組織を作成する場合は、ナビゲーションパネルを使用して、サブ組織を作成する組織を選択します。
3. ナビゲーションパネルで「新規」をクリックします。

「新規組織」ページがデータパネルに表示されます。
4. 「新規組織」ページで、組織またはサブ組織の名の値を入力します。
5. 「アクティブ」または「非アクティブ」の状態を選択します。

デフォルトは「アクティブ」です。状態は、組織またはサブ組織が存在する間、プロパティ矢印を選択していつでも変更できます。「非アクティブ」を選択すると、組織またはサブ組織へのログインが無効になります。
6. 「了解」をクリックします。

新規の組織またはサブ組織がナビゲーションパネルで表示されます。
7. 「表示」メニューから「サービス」を選びます。
8. 「新規」をクリックします。
9. 新しい組織のデスクトップサービスを有効にします。
 - a. ロケーションパネルで「アイデンティティ管理」を選択します。
 - b. 「表示」メニューから「組織」を選択します。
 - c. 新規作成された組織を選択します。
 - d. 「表示」メニューから「サービス」を選択します。
 - e. 「ポータルデスクトップ」を選択します。

- f. 「デフォルトチャンネル名」の値を「DummyChannel」から「JSPTabContainer」(または新しい組織で使用する最上位コンテナの名前)に変更します。
- g. 「ポータルデスクトップのタイプ」の値をデフォルトから「sampleportal」(または新しい組織で使用するデスクトップタイプ)に変更します。

サービスを追加する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. サービスを追加する組織またはサブ組織に移動します。
ナビゲーションパネルの「表示」メニューを使用します。
3. 「表示」メニューから「サービス」を選びます。
4. 「新規」をクリックします。
5. データパネルから追加する1つまたは複数のサービスを選択して、「了解」を選択します。

サービスのテンプレートを作成する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 追加したサービスがある組織またはサブ組織に移動します。

ナビゲーションパネルの「表示」メニューを使用します。

3. 「表示」メニューから「サービス」を選びます。
4. 追加したサービスの隣にあるプロパティ矢印をクリックします。
5. サービスのデフォルトの属性値を使用するか、変更し、「保存」をクリックします。

注 LDAP およびポリシー設定サービスでは、空白のパスワードフィールドは、root ユーザーバインド DN (cn=amldapuser,...) の下に位置します。このフィールドにはパスワードを入力して保存し、ポリシーおよび ldap を正しく設定する必要があります。このパスワードは、管理ユーザーパスワードとは異なります。これらのパスワードについては、UNIX 管理者に問い合せてください。

Access Manager 固有のサービス属性の設定については、『Access Manager 管理ガイド』を参照してください。Portal Server 固有のサービス属性の設定についての詳細は、このマニュアルの該当する付録を参照してください。

新規ユーザーを追加する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. ユーザーの作成先である組織またはサブ組織に移動します。
3. 「表示」メニューから「ユーザー」を選択し、「新規」をクリックします。

「新規ユーザー」ページがデータパネルに表示されます。

注 ドロップダウンメニューに「ユーザー」ではなく「ピープルコンテナ」が表示される場合は、「ピープルコンテナを表示」属性を組織に対して設定しているか、または最上位の特定のポイントで設定していることを確認します。これは、Access Manager サービスの「管理」で設定されています。

ユーザーは常にピープルコンテナに格納されますが、「ピープルコンテナを表示」属性を選択していない場合は、組織の1つ下にユーザーが表示されるため、ユーザーを操作できます。「ピープルコンテナを表示」はデフォルトでは設定されていません。

4. ユーザーに割り当てるサービスを選択して、「次へ」をクリックします。
一般に、ほとんどのユーザーには少なくともポータルデスクトップ、認証設定、登録の各サービスを追加します。
5. ユーザー情報を入力して、「終了」をクリックします。
新規ユーザーがナビゲーションパネルに表示されます。

ユーザーにサービスを追加する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. ユーザーの作成先である組織またはサブ組織に移動します。
3. 「表示」メニューから「ユーザー」を選択します。
4. ナビゲーションパネルでユーザーを選択し、プロパティの矢印をクリックします。
5. 「表示」メニューから「サービス」を選択します。
6. 「新規」をクリックして、ユーザーに割り当てるサービスを選択します。
7. サービスにチェックマークを付け、「了解」をクリックします。
一般に、ほとんどのユーザーには少なくともポータルデスクトップ、登録の各サービスを追加します。

新規ロールを作成する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. ロールの作成先である組織またはサブ組織に移動します。
3. 「表示」メニューから「ロール」を選択し、「新規」をクリックします。
「新規ロール」ページがデータパネルに表示されます。
4. ロール情報(名前、説明、ロールタイプ、アクセス権)を入力し、「終了」をクリックします。
新規ロールがナビゲーションパネルに表示されます。

注 代理の管理用にカスタマイズされたロールを作成している場合は、あらかじめそのロールに ACI 権限を定義する必要があります。詳細については、[第7章「管理の委任の設定」](#)を参照してください。

ロールをユーザーに割り当てる

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. ロールの作成先である組織またはサブ組織に移動します。
3. 「表示」メニューから「ユーザー」を選択します。

4. ロールの割り当て先であるユーザーの隣にあるプロパティの矢印を選択します。
データパネルにユーザープロフィール情報が表示されます。
5. データパネルの「表示」メニューから「ルール」を選択します。
「ルールを追加」ページが表示されます。
6. 割り当てるロールの隣にあるチェックボックスにチェックマークを付けて、「保存」をクリックします。
このユーザーのロールを表示するボックス内が、割り当てられたロールで更新されます。
7. 「保存」をクリックし、変更内容を保存します。

既存のユーザーの Portal Server へのアクセスを有効化する

Access Manager の既存のインスタンスに Portal Server をインストールする場合、ユーザーは、Portal Server Desktop を使用するよう追加されていません。ユーザーが Desktop にアクセスできるようにするには、ユーザーを有効にする必要があります。次の手順を使用して、デフォルトの組織または別の組織のユーザーを有効にします。

デフォルト組織のユーザーを有効化する

開始する前に、いくつかの設定情報を取得する必要があります。設定の詳細が一部不明な場合は、`/var/sadm/pkg/SUNWps/pkginfo` ファイルからスクリプトを使用して情報を取り出すことができます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. `/var/sadm/pkg/SUNWps/pkginfo` ファイルから、次の情報を確認または取り出します。
 - ディレクトリマネージャの識別名 (`DS_DIRMGR_DN/` と指定)。デフォルト値は `cn=Directory Manager`。
 - ディレクトリマネージャのパスワード (`DS_DIRMGR_PASSWORD/` と指定)。
 - ディレクトリサーバーの完全修飾ドメイン名 (`DS_HOST/` と指定)。

- ディレクトリサーバーが実行するポート (*DS_PORT/* と指定)。デフォルト値は 389 です。
- ディレクトリツリーのルートサフィックス (*DS_ROOT_SUFFIX/* と指定)。デフォルト値は *dc=orgname,dc=com* (*dc=sun,dc=com* など) です。
- Portal Server インストールのデフォルトの組織 (*DS_DEFAULT_ORG/* と指定)。デフォルト値は *o=domain-name* です。
- Portal Server インストールのベースディレクトリ。デフォルト値は */opt* です。

設定情報が不明な場合は、次のスクリプトを実行して出力を参照し、この手順を完了するのに必要な情報を取得してください。

```
#####
# Get configuration from file
#####

GrabConfig() {
    GRABCONFIG_KEY=$1
    GRABCONFIG_FILE=$2
    GRABCONFIG_SEPARATOR=$3
    ANSWER_CONFIG=' $GREP "^$GRABCONFIG_KEY$GRABCONFIG_SEPARATOR"
$GRABCONFIG_FILE | $UNIQ | $SED -e
"s/$GRABCONFIG_KEY$GRABCONFIG_SEPARATOR/" | $SED -e "s/^ //" '
}

#####
# Get PS6 Settings
#####

GetPS6Settings() {
    if [ -f $PKGINFO ]; then
        # Ldap Settings
```

```
#
    GrabConfig "DS_HOST" $PKGINFO "="
    DS_HOST=$ANSWER_CONFIG
    echo "DS_HOST=$DS_HOST"
    GrabConfig "DS_PORT" $PKGINFO "="
    DS_PORT=$ANSWER_CONFIG
    echo "DS_PORT=$DS_PORT"

GrabConfig "DS_DIRMGR_DN" $PKGINFO "="
    DS_DIRMGR_DN=$ANSWER_CONFIG
    echo "DS_DIRMGR_DN=$DS_DIRMGR_DN"
    GrabConfig "DS_DIRMGR_PASSWORD" $PKGINFO "="
    DS_DIRMGR_PASSWORD=$ANSWER_CONFIG
    echo "DS_DIRMGR_PASSWORD=$DS_DIRMGR_PASSWORD"

#####
# Get PS6 Settings
#####

GetPS6Settings() {

    if [ -f $PKGINFO ]; then
```

```

# Ldap Settings
#
GrabConfig "DS_HOST" $PKGINFO "="
DS_HOST=$ANSWER_CONFIG
echo "DS_HOST=$DS_HOST"
GrabConfig "DS_PORT" $PKGINFO "="
DS_PORT=$ANSWER_CONFIG
echo "DS_PORT=$DS_PORT"
GrabConfig "DS_DIRMGR_DN" $PKGINFO "="
DS_DIRMGR_DN=$ANSWER_CONFIG
echo "DS_DIRMGR_DN=$DS_DIRMGR_DN"
GrabConfig "DS_DIRMGR_PASSWORD" $PKGINFO "="
DS_DIRMGR_PASSWORD=$ANSWER_CONFIG
echo "DS_DIRMGR_PASSWORD=$DS_DIRMGR_PASSWORD"

# Dsame Settings
#
GrabConfig "IDSAME_BASEDIR" $PKGINFO "="
IDSAME_BASEDIR=$ANSWER_CONFIG
echo "IDSAME_BASEDIR=$IDSAME_BASEDIR"

AMCONFIG="$${IDSAME_BASEDIR}/SUNWam/lib/AMConfig.properties"
if [ -f $AMCONFIG ]; then
    DS_ROOT_SUFFIX='$GREP "^com.ipplanet.am.rootsuffix=" $AMCONFIG |
$SED -e "s/com.ipplanet.am.rootsuffix=//"'
    echo "DS_ROOT_SUFFIX=$DS_ROOT_SUFFIX"
    DS_DEFAULT_ORG='$GREP "^com.ipplanet.am.defaultOrg=" $AMCONFIG | ¥
$SED -e "s/com.ipplanet.am.defaultOrg=//"'
    echo "DS_DEFAULT_ORG=$DS_DEFAULT_ORG"

```

```

else
    print "'$GETTEXT 'Error - Cannot find DSAME configuration file,
please verify PS6 installation.'"
    exit 1
fi
else
    print "'$GETTEXT 'Error - Cannot find SUNWps package information
files, please verify PS6 installation.'"
    exit 1
fi

```

2. ディレクトリを Access Manager ユーティリティのディレクトリに変更します。たとえば、ベースディレクトリが /opt の場合は、次のように入力します。

```
cd /AccessManager-base/SUNWam/bin
```

3. ディレクトリサーバーとデフォルトの組織のルートサフィックスが同じでない場合は、次のコマンドを実行します。

```
./ldapsearch -h /DS_HOST/ -p /DS_PORT/ -D /DS_DIRMGR_DN/ -w /DS_DIRMGR_PASSWORD/  
-b "ou=People,/DS_DEFAULT_ORG/,/DS_ROOT_SUFFIX/" "(uid=*)" dn | /usr/bin/sed  
's/^version.*//' > /tmp/.tmp_ldif_file1
```

4. ディレクトリサーバーとデフォルトの組織のルートサフィックスが同じである場合は、次のコマンドを実行します。

```
./ldapsearch -h /DS_HOST/ -p /DS_PORT/ -D /DS_DIRMGR_DN/ -w /DS_DIRMGR_PASSWORD/  
-b "ou=People,/DS_ROOT_SUFFIX/" "(uid=*)" dn | /usr/bin/sed 's/^version.*//' >  
/tmp/.tmp_ldif_file1
```

5. 次のコマンドを実行します。

```
grep "^dn" /tmp/.tmp_ldif_file1 | awk '{  
print $0  
print "changetype: modify"  
print "add: objectclass"  
print "objectclass: sunPortalDesktopPerson"  
print "objectclass:sunPortalNetmailPerson\n" }' >  
/tmp/.tmp_ldif_file2
```

6. 次のコマンドを実行します。

```
./ldapmodify -c -h DS_HOST -p DS_PORT ¥ -D DS_DIRMGR_DN -w
DS_DIRMGR_PASSWORD -f /tmp/.tmp_ldif_file2
```

7. すべての一時ファイルを削除します。

```
rm /tmp/.tmp_ldif_file1 /tmp/.tmp_ldif_file2
```

デフォルト以外の組織のユーザーを有効化する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. /var/sadm/pkg/SUNWps/pkginfo ファイルから、次の情報を確認または取り出します。
 - ディレクトリマネージャの識別名 (*DS_DIRMGR_DN/* と指定)。デフォルト値は `cn=Directory Manager` です。
 - ディレクトリマネージャのパスワード (*DS_DIRMGR_PASSWORD/* と指定)
 - ディレクトリサーバーの完全修飾ドメイン名 (*DS_HOST/* と指定)
 - ディレクトリサーバーが実行するポート (*DS_PORT/* と指定)。デフォルト値は 389 です。
 - ディレクトリツリーのルートサフィックス (*DS_ROOT_SUFFIX/* と指定)。デフォルト値は `dc=orgname,dc=com` (`dc=sun,dc=com` など) です。
 - ユーザーを更新する Portal Server インストールの組織 (*DS_ORG_TO_UPDATE/* と指定)。デフォルト値は "" です。
 - Portal Server インストールのベースディレクトリ。デフォルト値は /opt です。
2. 有効にする既存のユーザーを含む組織またはサブ組織のサービスを追加します。手順の詳細については、「[サービスを追加する](#)」を参照してください。
3. 追加するサービスごとにテンプレートを作成します。手順の詳細については、「[サービスのテンプレートを作成する](#)」を参照してください。
4. サービスごとにポリシーを作成して割り当てます。詳細については、「[ピアまたはサブ組織のポリシーサービスを追加する](#)」、「[ピアまたはサブ組織の参照ポリシーを作成する](#)」、および「[ピアまたはサブ組織の標準のポリシーを作成する](#)」を参照してください。
5. 認証ユーザーの組織からの転送先に URL を設定します。「[ログインユーザーをポータルデスクトップ URL に正しくリダイレクトする](#)」を参照してください。

6. ディレクトリを Access Manager ユーティリティのディレクトリに変更します。たとえば、ベースディレクトリが /opt の場合は、次のように入力します。

```
cd /AccessManager-base/SUNWam/bin
```

7. 組織内のユーザーまたは組織を有効にするには、次のどちらかを実行します。
- DS_ORG_TO_UPDATE/として定義された特定の組織内のユーザーだけを有効にするには、次のコマンドを使用します(1行として入力する)。

```
./ldapsearch -h /DS_HOST/ -p /DS_PORT/ -D /DS_DIRMGR_DN/ -w  
/DS_DIRMGR_PASSWORD/  
-b "ou=People,/DS_ORG_TO_UPDATE/,/DS_ROOT_SUFFIX/" "(uid=*)" dn |  
/usr/bin/sed 's/^version.*//' > /tmp/.tmp_ldif_file1
```

- すべての組織のユーザーを有効にするには、次のコマンドを使用します(1行として入力する)。

```
./ldapsearch -h /DS_HOST/ -p /DS_PORT/ -D /DS_DIRMGR_DN/ -w  
/DS_DIRMGR_PASSWORD/  
-b "/DS_ROOT_SUFFIX/" "(uid=*)" dn | /usr/bin/sed 's/^version.*//' >  
/tmp/.tmp_ldif_file1
```

8. 次のコマンドを実行します。

```
grep "^dn" /tmp/.tmp_ldif_file1 | awk '{  
print $0  
print "changetype: modify"  
print "add: objectclass"  
print "objectclass: sunPortalDesktopPerson"  
print "objectclass:sunPortalNetmailPerson\n" }' >  
/tmp/.tmp_ldif_file2
```

9. 次のコマンドを実行します。

```
./ldapmodify -c -h DS_HOST -p DS_PORT ¥ -D "DS_DIRMGR_DN" -w  
DS_DIRMGR_PASSWORD -f /tmp/.tmp_ldif_file2
```

10. すべての一時ファイルを削除します。

```
rm /tmp/.tmp_ldif_file1 /tmp/.tmp_ldif_file2
```

11. ディレクトリを Portal Server のユーティリティのディレクトリに変更します。

```
cd /AccessManager-base/SUNWps/bin
```

12. 次のコマンドを実行して、デフォルト以外の組織のディスプレイプロファイルをロードします。

```
./dpadmin modify -u
"uid=amadmin,ou=people,DS_DEFAULT_ORG,DS_ROOT_SUFFIX" -w
DS_DIRMGR_PASSWORD -d
"NON_DEFAULT_ORG,DS_DEFAULT_ORG,DS_ROOT_SUFFIX"
AccessManager-base/SUNWps/samples/desktop/dp-org.xml
```

13. 別の組織のユーザーを有効にするには、[手順7](#)から[手順13](#)までを繰り返します。

ポータル組織の迅速な新規作成

次のタスクでは、組織を新規作成し、ポータル用途用に有効にする手順を説明します。デフォルトでは、ログイン時にロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. 新しい組織を作成します。
 - a. 「表示」メニューから「組織」を選択します。
 - b. 「新規」をクリックします。
組織の作成ページがデータパネルに表示されます。
 - c. 新規組織名を入力します。「組織の状態」を「アクティブ」にする必要があります。「了解」をクリックします。
ナビゲーションページに、新規作成した組織が表示されます。
2. 新しい組織のサービスを追加します。
 - a. ナビゲーションパネルの「表示」メニューから「組織」を選択し、「組織」リストから新たに作成した組織をクリックします。
 - b. 「表示」メニューから「サービス」を選択します。
 - c. 「追加」をクリックします。
「サービスを追加」ページがデータパネルに表示されます。組織用に登録するサービスを選択します。少なくとも次のサービスを追加する必要があります。
 - コア
 - LDAP またはこの組織に利用する認証サービス
 - メンバーシップ

- ポータルデスクトップ

この手順の目的では、次も登録する必要があります。

- ポリシー設定
- 登録
- ユーザー管理

新規に追加されたサービスがナビゲーションパネルに表示されます。

- d. プロパティの矢印をクリックして各サービスを設定します。設定属性を変更するときは、「作成」をクリックします。Portal Server に固有の設定以外の属性については、『Sun Java System Access Manager 管理ガイド』を参照してください。

注 サブ組織は、親組織とは別にサービスを追加する必要があります。

3. 親組織から新しい組織へ、Desktop 参照ポリシーを作成します。

参照は、ルール内のリソースとして親組織を定義する必要があり、参照内の値として SubOrgReferral にサブ組織が含まれている必要があります。

- a. ロケーションパネルで「アイデンティティ管理」を選択します。
- b. 親組織を選択します。
- c. 「表示」メニューから「ポリシー」を選択します。
- d. 「新規」をクリックし、新しいポリシーを作成します。
ポリシーの作成ページがデータパネルに表示されます。
- e. 「ポリシータイプ」で「参照」を選択します。
- f. 「名前」には、SubOrgReferral_Desktop を入力します。「了解」をクリックします。
ポリシーが作成され、「ポリシー」の下に表示されます。
- g. 「SunOrgReferral_Desktop」の隣にあるプロパティの矢印をクリックします。
- h. データパネルの「表示」メニューから「ルール」を選択し、「新規」をクリックします。「ポータルデスクトップ」が選択されていることを確認し、「次へ」をクリックします。
- i. ポータルデスクトップのルールに対する名前を指定し、「終了」をクリックします。
- j. データパネルの「表示」メニューから「参照」を選択し、「追加」をクリックします。データパネルで「値」にサブ組織の名前が選択されていることを確認し、「作成」をクリックしてポリシーの設定を完了します。

4. 新しい組織の通常のポータルデスクトップポリシーを作成します。
 - a. サブ組織に移動します。
 - b. 「表示」メニューから「ポリシー」を選択します。
その組織のポリシーが表示されます。
 - c. ナビゲーションパネルで「新規」を選択します。「新規ポリシー」ページがデータパネルに表示されます。
 - d. 「ポリシータイプ」で「標準」が選択されていることを確認します。
 - e. ポリシー用の名前を入力します。
 - f. 「了解」をクリックします。
 - g. データパネルの「表示」メニューから「ルール」を選択し、「新規」をクリックします。「新規ルール」ページがデータパネルに表示されます。
 - h. ルールタイプを選択し、「ルールアクション」の下にある処理を選択します。「終了」をクリックします。
 - i. データパネルの「表示」メニューから「サブジェクト」を選択し、「新規」をクリックします。「サブジェクトを追加」ページがデータパネルに表示されます。
 - j. ポータルデスクトップポリシーが適用されるサブジェクトを選択し、「次へ」をクリックしてサブジェクトの設定を完了します。
 - k. 「終了」をクリックして、ポリシーの設定を完了します。
5. 新しい組織の新規ユーザーを作成します。
 - a. ロケーションパネルで「アイデンティティ管理」を選択します。
 - b. 「表示」メニューから「組織」を選択します。
 - c. 新規作成された組織を選択します。
 - d. 「表示」メニューから「ユーザー」を選択します。
 - e. 「新規」をクリックします。
 - f. ユーザー用に登録するサービスを選択します。
 - g. 「次へ」をクリックします。
 - h. テキストフィールドにユーザーの詳細を入力します。
 - i. 「終了」をクリックします。
6. 新しい組織のデスクトップサービスを有効にします。
 - a. ロケーションパネルで「アイデンティティ管理」を選択します。
 - b. 「表示」メニューから「組織」を選択します。

- c. 新規作成された組織を選択します。
 - d. 「表示」メニューから「サービス」を選択します。
 - e. 「ポータルデスクトップ」を選択します。
 - f. 「ポータルデスクトップのタイプ」の値をデフォルトから「sampleportal」(または新しい組織で使用するデスクトップタイプ)に変更します。
7. 新しい組織の Desktop にアクセスします。
- a. 管理コンソールからログアウトします。
 - b. ブラウザページを開き、次のように入力します。
`http://server:port/amserver/UI/login?org=neworg`
 ユーザーの Desktop が表示されます。

認証の設定

ここでは、Portal Server 認証の構成について説明します。Access Manager が、認証のフレームワークを提供します。認証は、ユーザーの ID を確認するプラグインモジュールを通じて実装されます。Access Manager には、コア認証モジュールをはじめ 7 つの異なる認証モジュールがあります。Access Manager 管理コンソールを使用して、デフォルト値を設定し、認証サービスを追加し、組織の認証テンプレートを作成し、サービスを有効化します。コア認証モジュールは認証のための設定全般を実行するため、特定の認証モジュールを設定する前に、コア認証モジュールを追加し、コア認証モジュールのテンプレートを組織別に作成しておく必要があります。

注 Sun ONE Identity Server 5.1 管理コンソールで提供された認証メニューの設定機能は、Sun Java System Access Manager リリースではサポートされなくなりました。有効な認証モジュールを選択できるリストを設定する場合は、Access Manager 管理コンソールを使用して、認証レベル属性が同じ値を持つように各認証モジュールを設定してください。認証モジュールの設定については、「[認証メニューを設定する](#)」を参照してください。

インストール中に、コア認証が追加され、デフォルトの組織にテンプレートが作成されます。また、インストールにより、次の認証モジュールが追加され、テンプレートが作成されます。

- LDAP: LDAP 認証により、ディレクトリツリーの検索ベース内にある有効なユーザーはすべて、Portal Server にログインできます。これにより自動的にユーザーが特定のロールに割り当てられます。

- **メンバーシップ**:メンバーシップ認証により、ユーザーはアカウントを作成し、管理者の補助がなくてもそのアカウントをパーソナライズできます。ユーザーはこの新しいアカウントを使用して、追加ユーザーとしてアクセスできます。

注 インストールは、コア、LDAP、およびメンバーシップモジュールからなる基本認証の実装を設定しますが、新しい組織を作成する場合や、外部LDAPディレクトリまたはIDプロバイダを認証する機能など追加の認証機能をセットアップする場合には、手動で認証を設定する必要があります。

認証モジュールを設定する手順の概要は次のとおりです。

1. それぞれの新しい組織にコア認証サービスを追加します。サービスを追加する手順については、「[サービスを追加する](#)」を参照してください。
2. コア認証サービスにテンプレートを作成します。サービスにテンプレートを作成する手順については、「[サービスのテンプレートを作成する](#)」を参照してください。
3. 各組織にサポートする認証サービスを追加します。サービスを追加する手順については、「[サービスを追加する](#)」を参照してください。
4. 認証サービスに組織でサポートするサービステンプレートを作成します。認証サービスにテンプレートを作成する手順については、「[サービスのテンプレートを作成する](#)」を参照してください。サービスの属性の設定については、『Access Manager 管理ガイド』の第5章「[認証オプション](#)」を参照してください。
5. 認証メニューを設定します。認証順序を設定する手順については、「[認証メニューを設定する](#)」を参照してください。
6. 認証サービスの使用順序を設定します。認証順序を設定する手順については、「[認証順序を設定する](#)」を参照してください。

認証レベルによる認証

各認証モジュールには、認証レベルを表す整数値を関連付けることができます。認証レベルを割り当てるには、「サービス設定」で認証モジュールのプロパティの矢印をクリックし、認証レベル属性の値を変更します。認証レベルが高いほど、1つまたは複数の認証モジュールで認証されたユーザーの信頼度は高く定義されます。

認証メニューを設定する

ユーザーは、特定の認証レベルの認証モジュールにアクセスできます。たとえば、次の構文を使用して、一ユーザーとしてログインできます。

```
http://hostname:port/deploy_uri/UI/Login?authlevel=auth_level_value
```

ユーザーが選択できる認証のメニューには、*auth_level_value* 以上の認証レベルが設定されたすべてのモジュールが表示されます。一致するモジュールが1つだけ見つかった場合は、その認証モジュールのログインページが直接表示されます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ログイン時にロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 認証を設定する組織またはサブ組織に移動します。
ナビゲーションパネルの「表示」メニューを使用します。
3. 「表示」メニューから「サービス」を選択します。
4. コアの隣にあるプロパティ矢印をクリックします。
5. 「組織」セクションの「組織認証モジュール」フィールドで適切な認証モジュールを選択して有効にします。
デフォルトでは、Portal Server のインストールにより、LDAP およびメンバーシップが有効になります。
6. 各認証モジュールの「デフォルトの認証レベル」に値を入力します (デフォルト値は 0)。
認証メニューに表示するには、各認証モジュールの値が同じである必要があります。
7. 「保存」をクリックします。

認証順序を設定する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。

デフォルトでは、ログイン時にロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 認証を設定する組織またはサブ組織に移動します。

ナビゲーションパネルの「表示」メニューを使用します。
3. 「表示」メニューから「サービス」を選択します。
4. コアの隣にあるプロパティ矢印をクリックします。
5. 「組織」セクションの「組織認証モジュール」フィールドで適切な認証モジュールを選択して有効にします。

デフォルトでは、Portal Server のインストールにより、LDAP およびメンバーシップが有効になります。
6. 各認証モジュールの「デフォルトの認証レベル」に値を入力します (デフォルト値は 0)。

認証メニューに表示するには、各認証モジュールの値が同じである必要があります。
7. 各認証モジュールの属性情報を指定するには、「組織認証設定」で「編集」を選択します。
 - a. 「追加」をクリックして、認証モジュールをメニューに追加します。
 - b. 認証メニューに表示される認証モジュールの順序を変更するときは、「並べ替え」をクリックします。
 - c. 「保存」をクリックし、属性情報を保存します。
8. 「保存」をクリックします。
9. 次の URL を使用して、管理サーバーにログインし、認証メニューが適切な選択肢と一緒に表示されることを確認します。

`http://host:port/amserver/UI/login`

これがデフォルトの組織認証ではない場合には、次の URL を使用して組織の認証メニューを確認します。

`http://host:port/amserver/UI/login?org=org_name`

外部ディレクトリに LDAP 認証を設定する

Portal Server をインストールすると、インストールプログラムによりディレクトリインスタンスへの LDAP 認証が自動的に設定されます。インストールプログラムにより、ローカルサーバーにディレクトリの内部インスタンスをインストールし、その内部ディレクトリに対する LDAP 認証を設定、またはディレクトリの既存の外部インスタンスへの LDAP 認証を設定できます。初期設定を行うと、外部 LDAP ディレクトリへの認証を設定するいくつかのシナリオが想定されます。たとえば、パフォーマンスまたはセキュリティ上の理由から、特定の組織の認証情報を専用の LDAP サーバーに分離することができます。

amadmin ユーザーを含む組織については、外部 LDAP に認証を設定しないでください。これは、amadmin ユーザーの認証を妨げ、管理コンソールからロックアウトされる可能性があるためです。誤って amadmin ユーザーを含む組織を設定してしまった場合は、amadmin の完全 DN を使用してログインし、LDAP テンプレートを修正する必要があります。amadmin DN は、AMConfig.properties ファイルの `com.sun.authentication.super.user` プロパティに一覧表示されます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 認証を設定する組織またはサブ組織に移動します。
ナビゲーションパネルの「表示」メニューを使用します。
3. 「表示」メニューから「サービス」を選びます。
4. 「Access Manager 設定」の「コア」の隣にあるプロパティの矢印をクリックします。
5. 「ダイナミックユーザプロファイル」メニューの「ダイナミック」を選択します。

6. 「Access Manager 設定」メニューで「LDAP」の隣にあるプロパティの矢印をクリックします。
7. サーバーに適切な LDAP 属性を設定します。次の例は、ポート 389 の LDAP サーバー `ds-sesta1.sesta.com` へのアクセスに検索開始位置 `ou=people,dc=sesta,dc=com` を設定し、root ユーザーによる `cn=root,ou=people,dc=sesta,dc=com` へのバインドを設定します。

 プライマリ LDAP サーバーとポート：`ds-sesta1.sesta.com:389`
 セカンダリ LDAP サーバーとポート：`ds-sesta1.sesta.com:389`
 ユーザー検索の開始 DN：`ou=people,dc=sesta,dc=com`
 root ユーザーバインド DN：`cn=root,ou=people,dc=sesta,dc=com`
 root ユーザーバインドパスワード：`root password`
 ユーザーネーミング属性：`uid`
 ユーザーエントリ検索属性：`employeenumber`
 ユーザー検索フィルタ：空白
 検索範囲：サブツリー
 LDAP サーバーに対する SSL を有効：オフ
 認証するユーザー DN を返す：オフ
 認証レベル：0
8. 「保存」をクリックします。

匿名認証の設定

Portal Server は、匿名認証の実装について次の 2 つの方法をサポートしています。

- 認証なしのユーザー ID の属性を使用する。Desktop URL にアクセスしているユーザーは、自動的に認証され、Desktop へのアクセス権を与えられます。
- 匿名ユーザーセッションを使用する。ユーザーは、認証メニューから「匿名」を選択して `anonymous` としてログインし、Desktop へのアクセス権を与えられます。

匿名認証をサポートするため、Portal Server インストールプログラムは、ユーザーアカウント `authlessanonymous` を作成して、次の 2 つのポータルデスクトップサービスのグローバル属性で、このユーザーについてのアクセスを設定します。

- 許可されている認証なしユーザー ID
- デフォルトの認証なしユーザー ID

Portal Server は、次の操作を実行できるという点において、認証なしと匿名認証の同時設定をサポートします。

1. 認証なしモードでも機能するように Desktop を設定します。
2. 選択肢の 1 つとして「匿名」が表示されるように、認証メニューを設定します。

3. ブラウザ A で Desktop にアクセスします。この場合認証なしモードでのアクセスになります。
4. ブラウザ B で `http://server/amserver/UI/login` にアクセスして「匿名」を選択し、Desktop を表示します。

この時点では、ブラウザ A では認証なしモードを使用し、ブラウザ B では匿名モードを使用しています。

Desktop にアクセスする方法は 2 通りです。一方の認証なしのアクセスは `/portal/dt` を直接参照する方法を用いますが、もう一方 (匿名) では間接的に `/amserver/login` を使用します。

Access Manager のメニューに匿名ログインのみを設定すると、Access Manager の「ログイン」メニューを省略することができます。

認証なしと匿名の両方の認証方式が同時にサポートされることはありません。このため、Access Manager セッションを開始せずに `/portal/dt` にアクセスする場合、次のどちらかだけが実行されます。

- a. Desktop は `/amserver/UI/login` にリダイレクトし、これによって自動的に匿名ログインが行われ、ユーザーが `/portal/dt` にリダイレクトされます。
- b. Desktop が認証なしモードで実行されます。

認証なしアクセスを使用する場合、匿名認証を無効にする必要はありません。ただし、上記の項目 a を実行する場合は、認証なしアクセスモードを無効にする必要があります。

匿名認証を設定する (匿名ユーザーセッション方式)

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 認証を設定する組織またはサブ組織に移動します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. ロケーションパネルで「サービス設定」を選択します。
4. ポータルデスクトップサービスの隣にあるプロパティの矢印をクリックします。

データパネルにポータルデスクトップの属性が表示されます。

5. 「許可されている認証なしユーザー ID」属性でリストに表示された値を選択し、「削除」を選択します。
6. 「デフォルトの認証なしユーザー ID」属性でリストに表示された値を選択し、「削除」を選択します。
7. 「保存」をクリックします。
8. ロケーションパネルで「アイデンティティ管理」を選択します。
9. 「表示」メニューから「組織」を選択します。
作成したすべての組織がナビゲーションパネルに表示されます。
10. 認証を設定する組織またはサブ組織に移動します。
ロケーションパネルの「表示」メニューを使用します。
11. 「表示」メニューから「サービス」を選択します。
12. 匿名サービスを追加して、設定します。
詳細については、「[サービスを追加する](#)」および「[サービスのテンプレートを作成する](#)」を参照してください。
13. 認証メニューに「匿名」を追加します。
詳細については、「[認証順序を設定する](#)」を参照してください。
14. anonymous ユーザーアカウントを作成します。
詳細については、「[新規ユーザーを追加する](#)」を参照してください。

匿名認証を設定する (認証なしアクセス)

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. デフォルトでは、ログイン時にロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
作成したすべての組織がナビゲーションパネルに表示されます。

3. 認証を設定する組織またはサブ組織に移動します。
ナビゲーションパネルの「表示」メニューを使用します。
4. パスワード `authlessanonymous` で `authlessanonymous` ユーザーアカウントを作成します。
詳細については、「[新規ユーザーを追加する](#)」を参照してください。
5. ロケーションパネルで「サービス設定」を選択します。
6. ナビゲーションパネルで「ポータルデスクトップ」を選択します。
7. 「許可されている認証なしユーザー ID」属性に `authlessanonymous` ユーザーの完全識別名を追加します。たとえば、次のようになります。
`uid=authlessanonymous, ou=People, dc=sesta, dc=com`
8. 「デフォルトの認証なしユーザー ID」属性で `authlessanonymous` ユーザーの完全識別名を指定します。
9. 「保存」をクリックします。

ブラウザを閉じて再起動し、新しく設定した認証なしのユーザー ID 方式を使用して Desktop にアクセスする必要があります。認証なしのユーザー ID 方式により、クエリー文字列でユーザーアカウントの UID を指定できます。デフォルトでは、認証なしの UID は `desktop.suid` です。プレフィックス `desktop` は、`desktopconfig.properties` ファイルの設定パラメータ `cookiePrefix` によって制御されます。たとえば、デフォルト組織 `sestat.com` から Desktop にアクセスする場合は次の URL を使用します。

```
http://server:port/portal/dt?desktop.suid=uid=authlessanonymous,
ou=People,dc=sesta,dc=com
```

注 ユーザーの使用言語以外のロケールを持つブラウザを使用してユーザーがログインすると、他のすべてのユーザーはログインプロンプトで同じロケールを共有します。

この問題には、いくつかの方法で対応できます。

- `dp-anon.xml` の `JSPTabContainer` で `refreshTime` の値を 0 に変更し、キャッシュを無効にする。
 - ロケールごとに 1 つの割合で複数の認証なしユーザーを指定し、ブラウザのロケールに基づいて適切なユーザーに認証なしデスクトップをリダイレクトする。
-

連携ユーザー用の Portal Server の設定

Sun Java System Portal Server ソフトウェアは、Liberty Alliance 仕様に準拠した連携 ID を持つユーザーをサポートしています。Liberty シングルサインオンの連携ユーザーは、Portal Server でパーソナライズされたデスクトップに 2 回目以降は認証なしでアクセスできます。

Liberty が有効な認証サービスについては、『Sun Java System Access Manager 管理ガイド』を参照してください。サービスプロバイダとして機能する Portal Server の設定例は、次の場所に用意されています。

`PortalServer-base/SUNWps/samples/liberty`

連携ユーザーを設定する

デフォルトでは、連携ユーザーはサービスプロバイダとして機能する Sun Java System Portal Server にアクセスする権限を持ちません。Portal Server は、連携ユーザーを次のように扱えます。

- Liberty シングルサインオンの連携ユーザーは、パーソナライズされたデスクトップにアクセスできます。
 - Liberty シングルサインオンではない連携ユーザーは、ID プロバイダの認証ページにリダイレクトされます。
1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。

デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。

2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルで「ポータルデスクトップ」を選択します。
4. 「連携を有効」をクリックします。
5. ホストプロバイダの ID を指定します。
6. 「保存」をクリックします。

連携ユーザーの認証なしアクセスを設定する

デフォルトでは、連携ユーザーは認証なしポータルデスクトップにアクセスする権限を持ちません。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 認証を設定する組織またはサブ組織に移動します。
ナビゲーションパネルの「表示」メニューを使用します。
3. ロケーションパネルで「サービス設定」を選択します。
4. ナビゲーションパネルで「ポータルデスクトップ」を選択します。
5. 「連携ユーザーの認証なしアクセスを無効」のチェックマークを外します。
6. 「保存」をクリックします。

認証なしアクセスの詳細については、「[匿名認証を設定する \(認証なしアクセス \)](#)」を参照してください。

UNIX 認証を設定する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 「アイデンティティ管理」の「表示」メニューで、「組織」を選択します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. ロケーションパネルで「サービス設定」を選択します。
4. ナビゲーションパネルの「UNIX」の隣にあるプロパティ矢印をクリックします（「Access Manager 設定」の下）。
5. サーバーに適切な UNIX 属性を設定します。
6. 「保存」をクリックします。
7. 認証を設定する組織またはサブ組織に移動します。
ナビゲーションパネルの「表示」メニューを使用します。

8. 「表示」メニューから「サービス」を選びます。
9. ナビゲーションパネルで「新規」をクリックします。
10. データパネルの「認証」の下の「コア」をクリックします。
11. データパネルの「組織認証モジュール」メニューから「UNIX」を選択します。
12. 「保存」をクリックします。

組織レベルの UNIX 認証を設定する

「[UNIX 認証を設定する](#)」で説明した UNIX 認証は、UNIX のグローバルな設定です。次に説明する手順は、組織レベルでの設定です。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. ブラウザの Web アドレスフィールドに `http://fullservername:port/amconsole` と入力し、管理者 (amadmin) として Sun Java System Access Manager 管理コンソールにログインします。
2. ログイン画面では、ユーザー ID として `amadmin` を入力し、インストール時に指定したパスワードを入力します。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
3. 「アイデンティティ管理」の「表示」メニューで、「組織」を選択します。
作成したすべての組織がナビゲーションパネルに表示されます。
4. 「表示」メニューから「サービス」を選びます。
5. 「追加」を選択します。
6. 右のパネルで「UNIX」にチェックマークを付けて、「了解」をクリックします。
7. 「UNIX」の隣にあるプロパティの矢印をクリックします。
8. 「サービステンプレートの作成 (UNIX)」パネルで「はい」を選択します。
9. サーバーに適切な UNIX 属性を設定します。
10. 「保存」をクリックします。
11. 「コア」の隣にあるプロパティの矢印をクリックします。
12. 認証メニューで「UNIX」を強調表示し、「保存」を選択します。

Portal Server によるポリシー管理の使用法の概要

ここでは、Access Manager のポリシー管理機能の使用法について説明します。ポリシーを作成、変更、削除する手順については、Access Manager のマニュアルを参照してください。

Access Manager のポリシーサービスにより、ルールを定義したり、リソースにアクセスしたりすることができます。ポリシーは、ロールまたは組織に基づき、権限の提供、あるいは制約の定義を可能にします。Portal Server には、次の3つのポリシーが内蔵されています。

- Portal Server ポータルデスクトップ実行機能：ユーザーが Desktop を表示できるようにします。
- Portal Server NetMail 実行機能：ユーザーが NetMail を実行できるようにします。

注 個々のポリシーの割り当てに関する説明については、[第8章「ポータルデスクトップサービスの管理」](#)および[第11章「NetMail サービスの管理」](#)を参照してください。

デフォルトでは、ポリシー設定サービスは自動的に最上位の組織に追加されます。サブ組織は、親組織とは別にポリシーサービスを追加する必要があります。作成したポリシーサービスは、すべての組織に追加する必要があります。ポリシーを使用するための手順の概要は、次のとおりです。

1. 組織のポリシーサービスを追加します。(インストールで指定した組織に自動的に追加)。サブ組織は、親サービスを継承しないため、サブ組織のポリシーサービスを追加する必要があります。詳細については、「[サービスを追加する](#)」を参照してください。
2. ピアまたはサブ組織の参照ポリシーを作成します。組織のポリシーに関する定義と決定を別の組織に委託することができます。または、リソースのポリシー定義を別のポリシー製品に委託することもできます。この、ポリシーの作成と評価に関するポリシー委任は、参照ポリシーによって制御されます。これは、ルールと参照自体から構成されます。リソースを必要としないアクションがポリシーサービスに含まれる場合、サブ組織の参照ポリシーを作成することはできません。詳細については、「[ピアまたはサブ組織の参照ポリシーを作成する](#)」を参照してください。
3. ピアまたはサブ組織の標準のポリシーを作成します。アクセス権の定義には標準のポリシーを使用します。標準のポリシーは、複数のルール、サブジェクト、条件から構成されます。詳細については、「[ピアまたはサブ組織の標準のポリシーを作成する](#)」を参照してください。

ピアまたはサブ組織のポリシーサービスを追加する

ピアまたはサブ組織は、親サービスを継承しないため、ピアまたはサブ組織のポリシーサービスを追加する必要があります。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。

デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 参照ポリシーを作成する組織またはサブ組織に移動します。

作成したすべての組織がナビゲーションパネルに表示されます。
3. ナビゲーションパネルの「表示」メニューから「組織」を選択し、「名前」メニューから適切な組織を選択します。
4. 「表示」メニューから「サービス」を選択します。
5. 「追加」をクリックします。

「サービスを追加」ページがデータパネルに表示されます。少なくとも以下のサービスのチェックボックスを選択し、「了解」をクリックします。
 - LDAP
 - メンバーシップ
 - ポリシー設定
 - ポータルデスクトップ
 - NetMail
新規に追加されたサービスがナビゲーションパネルに表示されます。
6. プロパティの矢印をクリックして各サービスを設定します。設定属性を変更するときは、「作成」をクリックします。Portal Server に固有の設定以外の属性については、『Sun Java System Access Manager 管理ガイド』を参照してください。

ピアまたはサブ組織の参照ポリシーを作成する

組織のポリシーに関する定義と決定を別の組織に委託することができます。この、ポリシーの作成と評価に関するポリシー委任は、参照ポリシーによって制御されます。これは、ルールと参照自体から構成されます。参照は、ルール内のリソースとして親組織を定義する必要があり、参照内の値として `SubOrgReferral` または `PeerOrgReferral` に組織名が含まれている必要があります。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 参照ポリシーを作成するのに使用する組織またはサブ組織に移動します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「表示」メニューから「ポリシー」を選択します。
4. 「新規」をクリックし、新しいポリシーを作成します。
ポリシーの作成ページがデータパネルに表示されます。
5. 「名前」には、`SubOrgReferral_organization` または `PeerOrgReferral_organization` を入力します。「ポリシータイプ」で「参照」が選択されていることを確認します。「了解」をクリックします。
6. 「サービス」からサービスのタイプを選択し、「次へ」をクリックします。
7. データパネルの「表示」メニューから「ルール」を選択し、「追加」をクリックします。「次へ」をクリックします。
「ルールを追加」テンプレートがデータパネルに表示されます。
8. 「ルール名」にルールの名前を入力し、「終了」をクリックします。
9. データパネルの「表示」メニューから「参照」を選択し、「追加」をクリックします。
「参照を追加」テンプレートがデータパネルに表示されます。
10. 「名前」に `SubOrgReferralName` と入力します。
データパネルで「値」にサブ組織の名前が選択されていることを確認し、「作成」をクリックしてポリシーの設定を完了します。

11. データパネルで「保存」をクリックします。

データが保存されると、「ポリシープロパティが保存されました。」というメッセージが表示されます。

ピアまたはサブ組織の標準のポリシーを作成する

アクセス権の定義には標準のポリシーを使用します。標準のポリシーは、複数のルール、サブジェクト、条件から構成されます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Management Server 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. ポリシーを作成する組織またはサブ組織に移動します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「表示」メニューから「ポリシー」を選択します。
その組織のポリシーが表示されます。
4. ナビゲーションパネルで「新規」を選択します。「新規ポリシー」ページがデータパネルに表示されます。
5. 「名前」には、`SubOrgNormal_organization` または `PeerOrgNormal_organization` を入力します。「ポリシータイプ」で「標準」が選択されていることを確認します。「了解」をクリックします。
6. 「サービス」メニューからサービスを選択し、「次へ」をクリックします。「ルール名」にルールの名前を入力します。適切なサービスの実行権限が与えられるように、適切なチェックボックスにチェックマークが付けられていることを確認します。
7. データパネルの「表示」メニューから「ルール」を選択し、「追加」をクリックします。「ルールを追加」テンプレートがデータパネルに表示されます。
8. データパネルの「表示」メニューから「サブジェクト」を選択し、「追加」をクリックします。「サブジェクトを追加」ページがデータパネルに表示されます。
9. 「終了」をクリックして、ポリシーの設定を完了します。

データが保存されると、「ポリシープロパティが保存されました」というメッセージが表示されます。

Portal Server デスクトップへのログイン

サンプルポータルをインストールしている場合は、ユーザーはサンプル Desktop にログインできます。また、Portal Server は、その他のユーザーにさまざまなログインをサポートしています。ここでは、Portal Server にログインできる、その他のいくつかの方法について説明します。

サンプルのポータルデスクトップにログインする

サンプル Desktop にアクセスするには、次の URL を入力します。

```
http://server:port/portal/dt
```

サブ組織にログインする

ユーザーに組織へのアクセス権がある場合、組織内のサブ組織にログインすることもできます。たとえば、ユーザーがサブ組織 B を持つ組織 A にアクセスできる場合、次の URL を入力してサブ組織 B にログインします。

```
http://server:port/amserver/UI/login?org=B
```

匿名認証を使用してログインする

注 匿名認証をサポートする場合、匿名認証モジュールを追加する必要があります。匿名認証モジュールを追加または有効にする詳細については、「[匿名認証の設定](#)」を参照してください。

1. 次の URL を使用してログインします。

```
http://server:port/portal/dt
```

2. Access Manager 認証ページで、「匿名」をクリックします。
3. サンプル Desktop が表示されます。

4. 必要に応じて、またメンバーシップ認証モジュールが追加されている場合、ログイン画面でユーザー ID を作成し、追加します。

ロギングの管理

Portal Server は、Access Manager ロギングおよびデバッグ API を使用します。

デフォルトでは、Portal Server のログファイルおよびデバッグファイルは、次のディレクトリに格納されています。

- /var/opt/SUNWam/logs
- /var/opt/SUNWam/debug

Access Manager 管理コンソールにより、次のロギングの属性を定義できます。

- ログの最大サイズ
- 履歴ファイルの数
- ログの場所
- ロギングタイプ
- データベースユーザー名
- データベースユーザーパスワード
- データベースドライバ名

詳細については『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

管理の委任の設定

この章では Sun Java™ System Portal Server の管理の委任を設定する方法について説明します。

この章で説明する内容は次のとおりです。

- [管理の委任の概要](#)
- [管理の委任モデルの定義](#)
- [管理の委任の設定](#)

管理の委任の概要

企業が作成するポータルが大規模かつ複雑になるに従い、中央集中型管理モデルはもはや実現不可能です。管理の委任、すなわち **Line of Business (LOB)** 管理は、管理作業を実際のポータルユーザーに委任または分散することで、この問題に対処しています。

Portal Server では、ロールを使用して管理機能をユーザーに委任できます。企業はロールベースの管理によって、規模の小さな組織または LOB に事業を分割できます。これによって異なるユーザーがユーザーのロールに基づいて LOB の組織、サブ組織、ユーザー、ポリシー、ロール、チャンネルを管理できます。

[144 ページの表 7-1](#) は、Portal Server で適用されるいくつかの重要な管理の委任のリストと定義を示しています。この表には 2 つの列があります。最初の列は用語を示し、2 番目の列は簡単な説明を示します。

表 7-1 管理の委任の用語

用語	説明
権限	単一のリソースと、そのリソースに基づいて実行できる単一のアクションの組み合わせです(たとえば、スタティックな Web ページの表示、給与アプリケーションへの支払い控えの表示、給与アプリケーションの W-4 データの修正など)。
アクション	あるリソースで実行できる手順または操作を表します(たとえば、カタログの読み出し、カタログの記述、POP による電子メールの入手、IMAP による電子メールの入手など)。
リソース	リソースはソフトウェアで抽象的に表現でき、アクセスが制御および保護されます。Sun Java System Access Manager では、リソースは URL アクセスだけを意味します。
Top-level Admin Role (最上位管理者ロール)	ポリシーと ID のすべての設定に対して完全な管理権限を持つロールです。
Organization Admin Role (組織管理者ロール)	特定の組織のポリシーと ID のすべての設定に対して完全な管理権限を持つロールです。
Line of Business (LOB)	LOB 機能はビジネス アナリストまたは同等の地位の者が実行できる管理機能です。LOB 管理者は最上位管理者機能を必要としない管理作業を実行できます。通常、リソースへのアクセス権を与えるロールにユーザーを追加する、またはそのロールからユーザーを削除するなどの LOB 機能は対象範囲内でのみ使用できます。
ロール管理者ロール	その他の特定のロールと特定のユーザーオブジェクトセットを管理するためのアクセス権を持つロールです。たとえば、ロールへのユーザーの追加と削除、またはロールレベルの属性の編集などを実行できます。
ロール管理者	ロール管理者ロールが割り当てられたユーザーです。

委任管理ロール

Sun Java System Access Manager 管理コンソールは、ロールベースの管理の委任の機能を異なる種類の管理者に割り当て、指定されたアクセス権に基づいて組織、ユーザー、ポリシー、ロール、チャンネルを管理します。

Sun Java System Access Manager 管理コンソールには、管理機能を委任するために、管理者ロールがあらかじめいくつか設定されています。これらのロールは次のとおりです。

- Top-Level Admin
- Group Admin

- Organization Admin
- Organization Help Desk Admin
- People Container Admin
- Container Admin
- Container Help Desk Admin

これらのロールについての詳細は、Sun Java System Access Manager 製品のマニュアルを参照してください。

注 また、Sun Java System Access Manager はその他に 3 つのロールを実装しています。Top-level Admin、Top-level Help Desk Admin、および Deny Write Access です。これらのロールはインストール時に作成され、インストールのルートにのみ存在します。新たに作成された組織には、これらのロールはありません。デフォルトでは、組織が新規作成されると、3 つのロール、すなわち Organization Admin、Organization Help Desk Admin、および People Admin が作成されます。

これらの事前に定義された管理者ロールを使用して、ロールの機能がニーズに合う場合は委任管理の実装を設定できます。たとえば、1 つの組織の下に複数のサブ組織があるディレクトリ構造のモデルを使用している場合、Organization Admin を複数のユーザーに割り当て、サブ組織ごとに委任管理者を作成することができます。ただし、企業の組織構造がより複雑な場合は、特定のニーズに的を絞った管理の委任を作成する方がよい可能性があります。この場合、Sun Java System Access Manager 管理コンソールを使用して、特定の業務上のニーズに合わせて委任管理者ロールを定義できます。

企業に合った管理の委任モデルを実装する場合、3 つの重要な概念ロールがあります。

- Top-level Admin Role
- Organization Admin Role
- Role Administrator Role

Top-level Admin Role はシステムの設定時に作成され、Organization Admin Role は新しい組織の設定時に自動的に作成されます。Role Administrator Role は、委任管理モデルの要件に基づいて作成するロールです。Role Administrator Role のアクセス権は、対応するアクセス制御命令 (ACI) を直接編集して定義します。

管理の委任では、次の原則が適用されます。

- ユーザー権限はユーザーのロールによって与えられます。
- 権限は個々のユーザーに基づいて、要求される権限を持つロールを定義し、このロールを個々のユーザーに割り当てることで許可されます。

- 特定のロールを1つ割り当てることで、一連のユーザーをグループ化できます。これらのユーザーは権限セットが許可され、そのロールに定義されているダイナミックな属性の値を継承します。
- ユーザーは複数のロールまたは集約ロールを持つことができます。複数のロールを持つユーザーは、ロールの機能の組み合わせすべてにアクセスできます。集約されたロールで許可される機能に競合が起これば、それらのロールの各サービスに対して定義される競合の解決レベルを通じて設定された優先順位に基づき、競合が解決されます。最高から最低まで7段階で解決レベルを設定できます。複数のロールのロールテンプレートがマージされることによって属性の競合が発生した場合は、競合の解決レベルが最も高いテンプレートセットの属性が返されます。

管理の委任モデルの定義

Portal Server の管理機能を適切に委任するために、企業が要求する管理ロールを決定する際に役立つ、管理の委任モデルを開発する必要があります。モデルを開発する場合、次の点に注意してください。

- 企業のビジネス要件を絞り込みます。一般に、ロールベースの委任管理に提案されるソリューションは、ビジネス上の要件に沿ったものにする必要があります。
- ユーザーが必要なリソースにアクセスし、ユーザーの管理上のニーズを委任された管理者が管理できるように、ユーザーをグループ化するためのディレクトリ構造を開発します。
- すべての事業要件に対応しながら、事業エンティティを可能な限り標準的なツリー構造に合わせるように試みます。組織とサブ組織を階層化した構造、またはフラットなディレクトリツリー構造を使用できます。フラットなディレクトリ構造では、すべてのエンティティは最上位の組織の1つ下に定義され、すべてのロール (Role Administrator Roles を含む) は組織階層としては互いに同位になります。たとえば、事業部門に関連するすべてのユーザーは、最上位組織の下のユーザーコンテナに作成されます。モデルに必要なアクセスロールと管理ロールのそれぞれについて、トップレベルで対応するロールを作成します。

管理の委任の設定

Portal Server で管理の委任を設定するための手順の概要は、次のとおりです。

1. Role Administrator Roles の ACI 設定を定義します。
2. 委任モデルのための Admin Roles を新規作成します。
3. Role Administrator Roles をユーザーに割り当てます。
4. ロールへの追加制限を設定します。

Role Administrator Roles の ACI 設定の定義

委任モデルに特定したロール管理者ロールに適切な権限を設定するために、委任モデルの固有の各ロールについて、ACI で適切な権限を定義する必要があります。ロールの ACI 権限テンプレートは、Sun Java System Access Manager 管理コンソールまたは Directory Server コンソールを使用して定義します。ldapmodify コマンドを使用すると、特定のロールに対して ACI を定義できます。

Sun Java System Access Manager 管理コンソールまたは Directory Server コンソールで ACI 権限テンプレートを定義する場合は、次のフォーマットを使用します。

```
permission_name | aci_desc | dn:aci ## dn:aci ## dn:aci
```

各表記の意味は次のとおりです。

permission_name は権限の名前です。

aci_desc はこれらの ACI で許可されるアクセスの詳細です。

dn:aci は DN と ACI のペアを表し、## で区切られます。Sun Java System Access Manager は各 ACI を関連する DN エントリに設定します。

このフォーマットは、値を代入できるタグもサポートします。この値は ACI では ROLENAME、ORGANIZATION、GROUPNAME、および PCNAME といった文字で指定する必要があります。これらのタグを使用することで、デフォルトとして使用できる十分な柔軟性を備えたロールを定義できます。デフォルトロールに基づいてロールが作成されると、ACI のタグはロールの DN から引き出された値で解決されます。

ACI の設定については、『Sun Java System Access Manager Programmer's Guide』を参照してください。

注 これらの ACI 定義例では、ルートサフィックスが dc=sesta,dc=com であることを前提としています。

コマンド行を使用して ACI を定義する

1. ldapmodify コマンドで、使用する ACI 設定を保存したテキストファイルを作成します。たとえば、次の acis.ldif ファイルには、JDCAdmin1 および JDCAdmin2 という 2 つのロールの ACI 定義が格納されています。

```
dn:dc=sesta,dc=com
changetype:modify
# aci for JDCAdmin1 role
# This role can add/delete users from JDC role
add:aci
aci:(target="ldap:///ou=people,dc=sesta,dc=com") (targetattr = "*") (version
3.0; acl "Allow JDCAdmin1 Role to read and search users"; allow (read,search)
roledn = "ldap:///cn=JDCAdmin1,dc=sesta,dc=com");
-
add:aci
aci:(target="ldap:///dc=sesta,dc=com")
(targetfilter="(entrydn=cn=JDC,dc=sesta,dc=com)") (targetattr="*") (version 3.0;
acl "Allow JDCAdmin1 Role to read and search JDC Role";allow (read,search)
roledn="ldap:///cn=JDCAdmin1,dc=sesta,dc=com");
-
add:aci
aci:(target="ldap:///ou=people,dc=sesta,dc=com") (targetattr="nsroledn") (target
filter="(!(|(nsroledn=cn=Top-level Admin
Role,dc=sesta,dc=com)(nsroledn=cn=Top-level Help Desk Admin
Role,dc=sesta,dc=com)(nsroledn=cn=Organization Admin
Role,dc=sesta,dc=com)(nsroledn=cn=Top-level Policy Admin
Role,dc=sesta,dc=com)))") (targetattrfilters="add=nsroledn:(nsroledn=cn=JDC,dc=se
sta,dc=com),del=nsroledn:(nsroledn=cn=JDC,dc=sesta,dc=com)") (version 3.0; acl
"Allow JDCAdmin1 Role to add/remove users to JDC Role"; allow
(write)roledn="ldap:///cn=JDCAdmin1,dc=sesta,dc=com");
-
# aci for JDCAdmin2 role
# This role can add/remove channels from the JDC role's display profile
add:aci
aci:(target="ldap:///cn=SunPortalDesktopService,dc=sesta,dc=com") (targetfilter
=(cn=cn=JDC,dc=sesta,dc=com)) (targetattr="*") (version 3.0; acl "Allow
JDCAdmin2 to edit display profile of JDC Role"; allow (all)
roledn="ldap:///cn=JDCAdmin2,dc=sesta,dc=com");
-
add:aci
aci:(target="ldap:///dc=sesta,dc=com") (targetattr = "*") (version 3.0; acl
"Allow JDCAdmin2 to read and search all"; allow (read,search) roledn =
"ldap:///cn=JDCAdmin2,dc=sesta,dc=com");)
```

2. ディレクトリを Sun Java System Access Manager ユーティリティのディレクトリに変更します。たとえば次のように指定します。

```
cd /AccessManager-base/SUNWam/bin
```

3. AccessManager-base/SUNWam/ldaplib/solaris/sparc/ldapsdk が含まれるように LD_LIBRARY_PATH を設定します。

4. 次のコマンドを実行します。

```
./ldapmodify -D "DS_DIRMGR_DN" -w DS_DIRMGR_PASSWORD -f /tmp/acis.ldif
```

5. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。

デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。

6. 新しいロールを作成する組織またはサブ組織に移動します (JDCAdmin1、JDCAdmin2 など)。

- a. 「表示」メニューから「ロール」を選択し、「新規」をクリックします。

- b. 「新規ロール」ページがデータパネルに表示されます。

- c. ロールの情報 (名前、説明、ロールのタイプ、アクセス権) を入力し、「終了」をクリックします (たとえば、タイプがサービスでアクセス権を持たないスタティックなロールである JDC、など)。

新規ロールがナビゲーションパネルに表示されます。

7. 作成したロールの「デスクトップ」サービステンプレートを作成します。

- a. 「表示」メニューから「サービス」を選びます。

- b. デスクトップサービスの隣にあるプロパティの矢印をクリックします。

- c. デスクトップサービスのデフォルトの属性値を使用するか、あるいは変更し、「保存」をクリックします。

8. ロールディスプレイプロファイルにタブを作成します (たとえば、JDC のロールディスプレイプロファイル、など)。

- a. タブを作成するロールに移動します。

- a. ナビゲーションパネルの「表示」メニューから「サービス」を選択します。

- b. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティ矢印を選択します。

- c. デスクトップ属性ページがデータパネルに表示されます。

- d. デスクトップページで「チャンネルおよびコンテナを管理」リンクを選択します。

- e. 「チャンネル」ページが表示され、コンテナパスが **root** で設定されています。
- f. チャンネルまたはコンテナを追加するコンテナをクリックします。
- g. ページの上部に、チャンネルが追加されるコンテナパスが表示されます。チャンネルとコンテナが定義されている場合は、リストに表示されます。
- h. 「追加」をクリックして、コンテナチャンネルまたはチャンネルを追加します。
- i. コンテナチャンネルを追加するには、「コンテナチャンネル」で「追加」をクリックします。チャンネルを追加するには、「チャンネル」で「追加」をクリックします。
- j. 「チャンネルの追加」ページが表示されます。
- k. チャンネル名を入力し、メニューからプロバイダのタイプを選択します。
- l. 「作成」をクリックします。

詳細については、[第 10 章「ディスプレイプロファイルの管理」](#)を参照してください。

9. ユーザーを作成します (admin1、admin2 など)。
 - a. ユーザーを作成するロールに移動します。
 - b. 「表示」メニューから「ユーザー」を選択し、「新規」をクリックします。
 - c. 「新規ユーザー」ページがデータパネルに表示されます。
 - d. ユーザーに割り当てるサービスを選択して、「次へ」をクリックします。
 - e. ユーザー情報を入力して、「作成」をクリックします。
 - f. 新規ユーザーがナビゲーションパネルに表示されます。
10. ユーザーにロールを割り当てます (たとえば、admin1 に JDCadmin1 を割り当てる、admin2 に JDCadmin2 を割り当てる、など)。
 - a. ロールの割り当て先である組織またはサブ組織に移動します。
 - b. 「表示」メニューから「ユーザー」を選択します。
 - c. ロールの割り当て先であるユーザーの隣にあるプロパティの矢印を選択します。
 - d. データパネルにユーザープロファイル情報が表示されます。
 - e. データパネルの「表示」メニューから「ロール」を選択します。
 - f. 「ロールを追加」ページが表示されます。
 - g. 割り当てるロールの隣にあるチェックボックスにチェックマークを付けて、「保存」をクリックします。
 - h. このユーザーのロールを表示するボックス内が、割り当てられたロールで更新されます。

- i. 「保存」をクリックし、変更内容を保存します。
11. 管理コンソールからログアウトします。

管理コンソールを使用して ACI を定義する

1. 最上位管理者として Sun Java System Access Manager 管理コンソールにログインします。

デフォルトでは、ロケーションメニューでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。

2. ロケーションパネルで「サービス設定」をクリックします。
3. 「管理」サービスの隣にあるプロパティの矢印をクリックします。

データパネルに管理の属性が表示されます。

4. 「デフォルトロールアクセス権 (ACI)」入力フィールドに、ACI 定義を入力して「追加」をクリックします。たとえば、以前に定義された JDCAdmin1 と JDCAdmin2 というロールの場合、次のように入力します。

```
JDCAdmin1|Add/delete users from JDC role|dc=sesta,dc=com:aci:(target=
"ldap:///ou=people,dc=sesta,dc=com") (targetattr = "*") (version 3.0; acl
"Allow JDCAdmin1 Role to read and search users"; allow (read,search)
roledn =
"ldap:///cn=JDCAdmin1,dc=sesta,dc=com");##dc=sesta,dc=com:aci:(target="
ldap:///dc=sesta,dc=com")
(targetfilter="(entrydn=cn=JDC,dc=sesta,dc=com)") (targetattr="*") (versi
on 3.0; acl "Allow JDCAdmin1 Role to read and search JDC Role";allow
(read,search) roledn="ldap:///cn=JDCAdmin1,dc=sesta,dc=com");
##dc=sesta,dc=com:aci:(target="ldap:///ou=people,dc=sesta,dc=com") (targ
etattr="nsroledn") (targetfilter="(!(|(nsroledn=cn=Top-level Admin
Role,dc=sesta,dc=com)(nsroledn=cn=Top-level Help Desk Admin
Role,dc=sesta,dc=com)(nsroledn=cn=Organization Admin
Role,dc=sesta,dc=com)(nsroledn=cn=Top-level Policy Admin
Role,dc=sesta,dc=com)))") (targetattrfilters="add=nsroledn:(nsroledn=cn=JD
C,dc=sesta,dc=com),del=nsroledn:(nsroledn=cn=JDC,dc=sesta,dc=com)") (ver
sion 3.0; acl "Allow JDCAdmin1 Role to add/remove users to JDC Role";
allow (write)roledn="ldap:///cn=JDCAdmin1,dc=sesta,dc=com");
```

```
JDCAdmin2|Add/remove channels from the JDC
role|dc=sesta,dc=com:aci:(target="ldap:///cn=SunPortalDesktopService,dc
=sesta,dc=com") (targetfilter=(cn=cn=JDC,dc=sesta,dc=com)) (targetattr="*
") (version 3.0; acl "Allow JDCAdmin2 to edit display profile of JDC
Role"; allow (all)
```

```
roledn="ldap:///cn=JDCAdmin2,dc=sesta,dc=com");##dc=sesta,dc=com:aci:(target="ldap:///dc=sesta,dc=com")(targetattr = "*" ) (version 3.0; acl "Allow JDCAdmin2 to read and search all"; allow (read,search) roledn = "ldap:///cn=JDCAdmin2,dc=sesta,dc=com");)
```

「デフォルトロールアクセス権 (ACI)」リストに新しい ACI が表示されます。

5. 「保存」をクリックします。

委任モデルのための管理ロールの新規作成

委任管理ロールの権限を定義する ACI を作成した後、その ACI 定義を使用するためのロールを作成する必要があります。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. 最上位管理者または組織管理者として Sun Java System Access Manager 管理コンソールにログインします。

デフォルトでは、ロケーションメニューでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。

2. ロールの作成先である組織またはサブ組織に移動します。

作成したすべての組織がナビゲーションパネルに表示されます。

注 これが新規組織の場合、すべてのサービスを追加し、適切なテンプレートを作成する必要があります。詳細については、[第 6 章「認証、ユーザー、およびサービスの管理」](#)を参照してください。

3. 「表示」メニューから「ロール」を選択し、「新規」をクリックします。

「新規ロール」ページがデータパネルに表示されます。

4. 名前を入力し、スタティックロールを選択して「次へ」をクリックします。

5. 説明を入力し、タイプとして「管理」を選択します。

6. 「アクセス権」を選択します。

- a. 管理コンソールを使用してロールの ACI 定義を作成する場合、「アクセス権」リストから作成したロールを選択します。

- b. コマンドラインを使用してロールの ACI 定義を作成する場合、「アクセス権」リストにロール名が表示されていないため「アクセス権なし」を選択します。

7. 「作成」をクリックします。
新規ロールがナビゲーションパネルに表示されます。

ロール管理者ロールを割り当てる

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションメニューでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. ロールの作成先である組織またはサブ組織に移動します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「表示」メニューから「ロール」を選択します。
4. 割り当てるロールのプロパティの矢印をクリックします。
5. データパネルの「表示」メニューから「ユーザー」を選択し、「追加」をクリックします。
「ユーザーを追加」ページがデータパネルに表示されます。
6. 割り当てるユーザーを検索するためのフィールドの値を指定し、「フィルタ」を選択します。
ユーザーのリストが表示されます。
7. ロールを割り当てるユーザーの隣のボックスにチェックマークを付けるか、「すべて選択」を選択してすべてのユーザーを選択します。
8. 「送信」をクリックします。
「このロールのユーザー」ボックスのリストが、割り当てられたユーザーで更新されます。

ロール管理者ロールで追加制限を設定する

機能が制限されたロールを設定できます。一般的な制限として、ディスプレイプロファイルの修正およびコンテンツ管理機能の実行に対するロールへの制限がありますが、この場合、デスクトップ属性の他の表示が制限されます。

また、委任された管理者を開始 DN ビューとともに設定することもできます。開始 DN ビューは、委任された管理者がエンティティを確認して修正できるディレクトリの場所です。

ロールに追加制限を設定するには、次の手順に従います。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションメニューでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 設定するロールが格納された組織またはサブ組織に移動します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「表示」メニューから「ロール」を選択します。
4. 設定するロールを選択します。
5. 「表示」メニューから「サービス」を選択します。
6. プロファイルまたはチャンネル管理機能のみを表示するようにロールを制限するには、次の手順に従います。
 - a. デスクトップサービスの「編集」リンクをクリックします。
 - b. このロールでユーザーサービステンプレートを作成します。
「デスクトップ」ページがデータパネルに表示されます。
 - c. 「ポータルデスクトップサービスの属性を表示」チェックボックスの選択を解除します。
 - d. 「管理者 DN 開始表示」に DN を指定します。
 - e. 「保存」をクリックします。

注 「ポータルデスクトップサービスの属性を表示」チェックボックスの選択が解除されている場合、このロールが割り当てられたユーザーが「デスクトップ」サービスにアクセスしても「デスクトップ」属性が表示されません。「チャンネルおよびコンテンツを管理」リンクが表示されるだけです。さらに、ロールレベルで定義されたチャンネルとコンテンツのみが表示されません。

7. 特定の開始 DN にロールを制限するには、次の手順を実行します。

- a. ユーザーサービスの「編集」リンクをクリックします。
- b. このロール用のユーザーサービステンプレートを作成します。
「ユーザー」ページがデータパネルに表示されます。
- c. 「管理者 DN 開始表示」に DN を指定します。たとえば、cn=JDC, dc=sesta, dc=com のように指定します。
- d. 「保存」をクリックします。

ポータルデスクトップサービスの管理

この章では、Sun Java System Portal Server デスクトップサービスの管理方法について説明します。

この章で説明する内容は次のとおりです。

- [デスクトップの概要](#)
- [チャンネルのホット配備の概要](#)
- [プロバイダアーカイブの概要](#)
- [ポータルデスクトップサービスの管理](#)
- [ポートレットの管理](#)
- [par ファイルの管理](#)

デスクトップの概要

ここでは、デスクトップコンポーネントとその基本構造、およびその管理方法について説明します。

デスクトップ用語集

[表 8-1](#) は、デスクトップに関連する用語の説明を示しています。

最初の列は用語を示し、2 番目の列は用語の定義を示します。

表 8-1 デスクトップ用語集

用語	定義
デスクトップ	Portal Server にプライマリエンドユーザーインタフェースを提供します。
プロバイダ	Portal Server が使用する汎用リソースのインタフェースとして機能します。JSP プロバイダは、JSP ファイルをコンパイルおよび実行し、マークアップを生成します。XML プロバイダは XML ファイルを変換してマークアップを生成します。Portal Server は、ポータルページで表示するマークアップに関する情報をプロバイダに照会できます。
ポートレット	ポータルのコンテキストの範囲内で要求を処理し、コンテンツを生成する、プラグイン可能な Web コンポーネント。ポートレットは、ポートレットコンテナ (JSR168 エクスポートグループによって定義されるポートレット仕様の実装) によって管理されます。概念的にはソフトウェアプロバイダと同じです。
チャンネル	デスクトップにコンテンツを表示します。通常は行と列で編成します。実行時のチャンネルは、チャンネルのサポートに必要なプロバイダオブジェクト、設定ファイル、およびすべてのデータファイル (JSP、HTML テンプレートなど) から構成されます。
コンテナまたはコンテナチャンネル	主に、他のチャンネル (子チャンネルと呼ばれる) のコンテンツを含めたり、集約したりすることでコンテンツを生成するチャンネルです。

ポータルデスクトップのアーキテクチャとコンテナ階層

デスクトップは、Portal Server のプライマリエンドユーザーインタフェースです。デスクトップはサブレットで実装され、さまざまな API とユーティリティ (Sun Java™ System Access Manager API、リソースバンドル、プロパティファイル、およびメールサーバーを始めとするバックエンドサーバーなど) によってサポートされています。

デスクトップは、プロバイダアプリケーションプログラミングインタフェース (PAPI) を使用してコンテンツを拡張、集約するメカニズムを提供します。コンテンツプロバイダ、またはプロバイダは、コンテナ階層、および一部のチャンネルを構築するための基本的なビルディングブロックを有効にします。通常、チャンネルは行と列で編成しま

すが、コンテナチャンネルを実装することで、この配備を変更することもできます。プロバイダは、コンテンツを生成するプログラムエンティティで、チャンネル中表示されます。生成したコンテンツは、ページ全体、フレーム、またはチャンネルなどの任意のマークアップで構成できます。

ポータルコンテンツ量が増加した場合にも、ポータルの構成、開発、および一般ユーザーによる使用実績は、参照するコンテンツのグループの格納方法によって簡略化できます。Portal Server は、コンテンツの集約においてコンテナプロバイダの柔軟で拡張性の高いセットを提供します。

図 8-1 は、デスクトップコンテナ階層の例を示しています。この図では、タブコンテナが最上位コンテナです。タブコンテナには、2つのタブチャンネルであるタブ 1 とタブ 2 があります。タブ 1 はテーブルコンテナで、内部に5つのチャンネルを格納します。

図 8-1 ポータルデスクトップのコンテナ階層の例

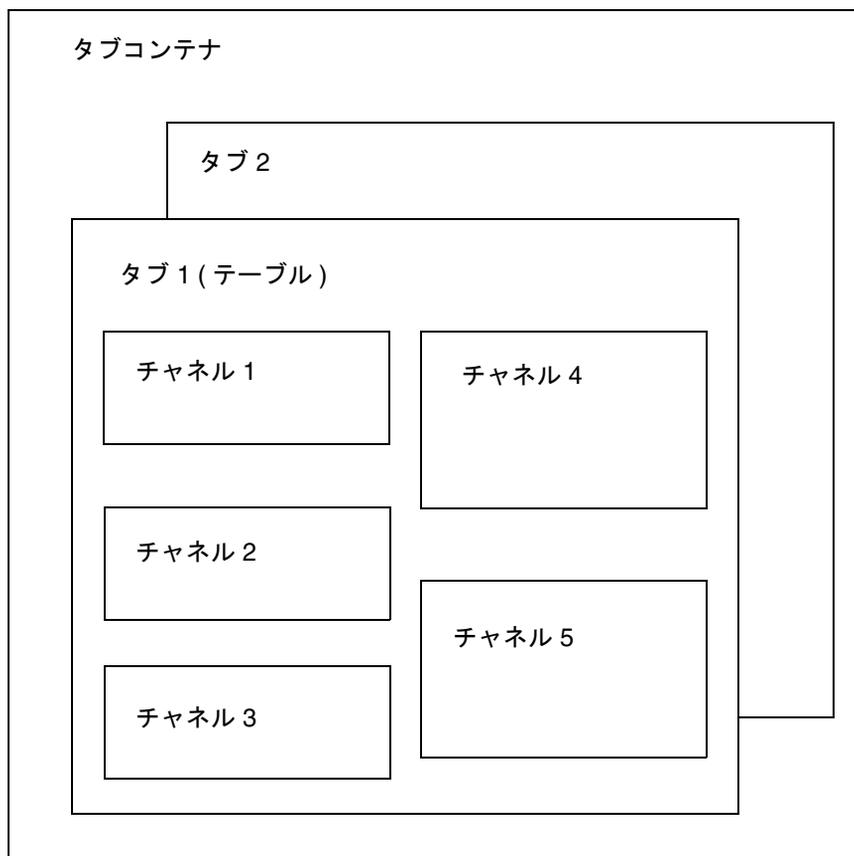


図 8-1 は、次の格納方式を示しています。

- **タブコンテナ** : 複数のテーブル、単一コンテナ、またはタブコンテナを格納します。このコンテナには、バナーおよびポータルのためのメニューバーも格納されます。
- **タブチャンネル** : チャンネルを切り替えるためにタブ付きのユーザーインターフェースを提供して、他のチャンネルの出力を集約します。実行時にタブコンテナの構成を修正して、表示するリーフチャンネルを変更できます。
- **テーブルコンテナ** : 他のチャンネルのコンテンツを行と列に集約します。このコンテナの機能は、Sun Portal Server 3.0 フロントプロバイダに相当します。このコンテナは、他チャンネルのコンテンツの入れ物と考えることができます。

ユーザー定義のチャンネル

それぞれのタブコンテナにはコンテンツリンクが含まれます。コンテンツリンクを選択すると、ユーザーが現在のタブのコンテナに表示するチャンネルを選択できるページが表示されます。このリリースでは、このページの右上に「新しいチャンネルの作成」というリンクが追加されています。「新しいチャンネルの作成」リンクを選択すると、ユーザーが新しいチャンネルを作成するためのページが表示されます。ただし、ユーザーが作成できるチャンネルは管理者が定義することができます。

表示されるページで新しいチャンネルを作成するには、ユーザーは [160 ページの表 8-2](#) に示される情報をフォームに指定する必要があります。

表 8-2 ユーザー定義のチャンネル

フォームのフィールド	フィールドのタイプ	フィールドの説明
チャンネル名	テキストフィールド	チャンネル名には英数字 (a ~ z、A ~ Z、0 ~ 9) だけを使用できます。
チャンネルのタイトル	テキストフィールド	チャンネルのタイトルバーに表示されるタイトルです。
チャンネルの説明	テキストフィールド	コンテンツリンクページに表示されるチャンネルの説明です。
チャンネルのタイプ	コンボボックス	新しいチャンネルの作成に使用できるプロバイダのリストです。
チャンネルのカテゴリ	コンボボックス	タブのコンテナのカテゴリのリストです。

表 8-2 ユーザー定義のチャンネル (続き)

フォームのフィールド	フィールドのタイプ	フィールドの説明
チャンネルを表示	「する」、「しない」のラジオボタン	「作成」ボタンの選択後にブラウザが更新されたときに、新しいチャンネルが自動的に表示されるようにする場合は、「チャンネルを表示」の「する」を選択します。「作成」ボタンの選択後にブラウザが更新されたときに、新しいチャンネルが自動的に表示されないようにする場合は、「しない」を選択します。この場合、コンテンツリンクからチャンネルを選択することで、チャンネルをブラウザに表示できます。どちらの場合も、ブラウザで新しいチャンネルを選択、表示したときは、そのチャンネルのタイトルバーに表示される「編集」ボタンを選択して、プロパティを更新する必要があります。
作成	ボタン	新しいチャンネルを作成するときに選択します。
取消し	ボタン	ユーザーがデスクトップの表示に戻るときに選択します。

ユーザー定義のチャンネルを作成すると、コンテンツページに「チャンネルの削除」リンクが表示されるようになります。ユーザーがリンクをクリックすると、ユーザーが作成したすべてのチャンネルが表示され、チャンネルを削除することができます。

ポータルデスクトッププロバイダ

Sun Java System Portal Server は、次の 2 つのタイプのプロバイダを使用します。

- ビルディングブロックプロバイダ: 拡張可能なプロバイダで、そのインタフェースは公開されています。これらのプロバイダは、汎用リソース (JSP ファイルなど) に接続します。これらのプロバイダは、ポータルデスクトップに複数のチャンネルを作成できるため、プロバイダとチャンネルの関係は一对多となります。
- コンテンツプロバイダ: 拡張不可能なプロバイダで、表示する固有のデータセットを必要とします (たとえば、ブックマークプロバイダは特定のテンプレートとデータを必要とする)。この種類のプロバイダは、ビルディングブロックプロバイダとは異なります。

ポータルデスクトップは、コンテンツ、プロバイダ、ポートレット、チャンネルデータを格納するためにディスプレイプロファイルを使用します。詳細については、[第 10 章「ディスプレイプロファイルの管理」](#)を参照してください。

ポータルデスクトップサービス

デスクトップサービスは、Sun Java System Access Manager サービスを使用して、各組織またはサブ組織のアプリケーションとユーザー固有の属性を格納します。管理者は、次にディスプレイプロファイルポリシーを作成し、それをユーザーに割り当てます。管理者はまた、Sun Java System Access Manager 管理コンソールを使用して、デスクトップ属性を修正します。デスクトップ属性についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

サンプルデスクトップ

Portal Server では、サンプルデスクトップに次のチャンネルが含まれます。

- ブックマーク
- アプリケーション
- ユーザー情報
- 検索
- ノート
- メールチェック
- ログイン
- シンプル Web サービス
- 設定可能なシンプル Web サービス
- ポートレットサンプル
- コラボレーション

これらのチャンネルは、サンプルポータルに合わせてカスタマイズおよび設定されています。配備するには、ユーザーインターフェースの修正が必要になることがあります。

ポータルデスクトップのカスタマイズ

Portal Server を配備する場合、ポータルの開発と、カスタマイズが主な作業になります。まず、プロバイダ、チャンネル、およびコンテナチャンネルを作成、拡張してからオンラインヘルプを配備し、見た目と使い心地などを変更します。必要に応じて、サンプルデスクトップを、サイトのポータルをカスタマイズする際の出発点として使用できます。ポータルのカスタマイズの詳細については、『Sun Java System Portal Server 6 2005Q1 Desktop Customization Guide』を参照してください。

チャンネルのホット配備の概要

Portal Server を使用すると、再起動せずに稼働中のシステムでプロバイダとチャンネルを配備できます。すなわち「ホット配備」が可能です。この場合、ユーザーセッションに割り込みが起こりません。

ホット配備を実現する 3 つのテクノロジーは次のとおりです。

- プロバイダクラスローダ: プロバイダ、およびプロバイダで使用するクラスをリロードします。プロバイダクラスローダを正しく実行するには、すべてのクラス (または JAR ファイル) を定義済みのディレクトリに配置する必要があります。
- ディスプレイプロファイルの再表示: Sun Java System Access Manager 管理コンソール、dpadmin コマンドなどの外部ソースで変更する場合、メモリ内のデスクトップ設定、つまりディスプレイプロファイルを更新します。
- ポータルデスクトップテンプレートと JSP のリロード: 設定されたデスクトップタイプに適切なテンプレートと JSP ファイルを取得します。

プロバイダアーカイブの概要

par ユーティリティで、チャンネル、ポートレット、プロバイダ、および関連するすべてのファイルを Portal Server システムにパッケージ化したり、システムから移動したりすることができます。チャンネル、ポートレット、プロバイダは .par ファイル形式で格納されます。.par ファイルに含まれるファイルは次のとおりです。

- ディスプレイプロファイルドキュメント
- クラスファイル
- プロバイダリソースバンドルファイル (プロパティファイル)
- テンプレートおよび JSP ファイル
- スタティックなコンテンツファイル、すなわち HTML および画像ファイル

ポータルデスクトップサービスの管理

デスクトップは、ユーザーのディスプレイプロファイルのマージしたセットに含まれるすべてのドキュメントをさらにマージし、それを使用してユーザーのデスクトップを構成します。ディスプレイプロファイルのマージしたセットには、ユーザーに関連するすべてのディスプレイプロファイルドキュメントが含まれます。ディスプレイプロファイルは、Sun Java System Access Manager 組織ツリーの異なるレベルで定義されます。ツリーのさまざまなレベルのディスプレイプロファイルドキュメントをマージまたは結合して、ユーザーのディスプレイプロファイルを作成します。たとえば、ユーザーのディスプレイプロファイルを形成するユーザーのディスプレイプロファイルドキュメントは、ロールディスプレイプロファイルドキュメント (存在する場合)、組織のディスプレイプロファイルドキュメント、およびグローバルなディスプレイプロファイルドキュメントとマージされます。

デスクトップディスプレイプロファイルとその他の設定データは、Sun Java System Access Manager サービス管理フレームワーク内にあるポータルデスクトップサービスのサービス属性として定義されます。組織が Sun Java System Access Manager 管理コンソールでポータルデスクトップサービスを追加する場合、組織内のすべてのユーザーはユーザープロファイルのポータルデスクトップサービス属性を継承します。この場合、ポータルデスクトップはこれらの属性を照会し、情報をどのようにポータルデスクトップに集約して表示するかを決定します。

デフォルトでは、ポリシー設定サービスは自動的に最上位の組織に追加されます。サブ組織は、親組織とは別にポリシーサービスを追加する必要があります。作成したポリシーサービスは、すべての組織に追加する必要があります。

Sun Java System Access Manager 組織のユーザーに対してポータルデスクトップサービスを設定するために必要な操作の概要は、次のとおりです。

1. 組織のポリシーサービスを追加します。
2. ピアまたはサブ組織の参照ポリシーを作成します。
3. ピアまたはサブ組織の標準のポリシーを作成します。
4. デフォルトのリダイレクト先 URL を割り当てます。
5. デスクトップサービス属性をカスタマイズします。

注 サンプルポータルをインストールすると、インストーラはサンプルに必要なディスプレイプロファイル XML ファイルをすべてインストールします。Sun Java System Access Manager コンソール、またはコマンド行インタフェースを使用すると、プロファイルをカスタマイズできます。詳細は、[第 10 章「ディスプレイプロファイルの管理」](#)を参照してください。

デフォルトでは、ポリシー設定サービスは自動的に最上位の組織に追加されます。サブ組織は、親組織とは別にポリシーサービスを追加する必要があります。作成したポリシーサービスは、すべての組織に追加する必要があります。ポリシーを使用するための手順の概要は、次のとおりです。

1. 組織のポリシーサービスを追加します (インストールで指定した組織に自動的に追加)。サブ組織は、親サービスを継承しないため、サブ組織のポリシーサービスを追加する必要があります。詳細については、「[サブ組織のポリシーサービスを追加する](#)」を参照してください。
2. ピアまたはサブ組織の参照ポリシーを作成します。組織のポリシーに関する定義と決定を別の組織に委託することができます (または、リソースのポリシー定義を別のポリシー製品に委託)。この、ポリシーの作成と評価に関するポリシー委任は、参照ポリシーによって制御されます。これは、ルールと参照自体から構成されます。リソースを必要としないアクションがポリシーサービスに含まれる場合、サブ組織の参照ポリシーを作成することはできません。詳細については、「[サブ組織の参照ポリシーを作成する](#)」を参照してください。
3. ピアまたはサブ組織の標準のポリシーを作成します。アクセス権の定義には標準のポリシーを使用します。標準のポリシーは、複数のルール、サブジェクト、条件から構成されます。詳細については、「[サブ組織に標準のポリシーを作成する](#)」を参照してください。

サブ組織のポリシーサービスを追加する

サブ組織は、親サービスを継承しないため、サブ組織のポリシーサービスを追加する必要があります。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. デスクトップサービスを追加する組織を選択します。
3. ナビゲーションパネルの「表示」メニューから「サービス」を選択します。
4. ナビゲーションパネルで「追加」をクリックします。
使用できるサービスのリストがデータパネルに表示されます。

5. 「Portal Server 設定」の「ポータルデスクトップ」チェックボックスにチェックマークを付けて、「了解」をクリックします。
ナビゲーションパネルが、「Portal Server 設定」に追加したデスクトップサービスで更新されます。
6. ナビゲーションパネルの「表示」メニューから「サービス」を選択します。
7. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティの矢印を選択します。
8. このとき、データパネルのメッセージ区画に、デスクトップサービスにサービステンプレートを作成する必要があるかどうかを確認するメッセージが表示されます。メッセージ区画内の「はい」をクリックし、テンプレートを作成します。
9. ページを送信してテンプレートが作成されると、データパネルにデスクトップサービス属性とデフォルト値があればその一覧が表示されます。必要に応じて値を修正します。完了したら、「保存」を選択してサービステンプレートの最新の値を保存します。

新規に作成したサービステンプレートのディスプレイプロファイルは、「サービス設定」にある「ポータルデスクトップ」サービスの「ダイナミック」セクションに入力した値を取ります。それらの値が空白の場合、この新規テンプレートのディスプレイプロファイルも空白になります。

注 「競合の解決レベル」属性の値は、デフォルト値では「最も高い」に設定されています。追加したサービスに同じ優先順位でサービステンプレートを各レベル(組織、ロールなど)でセットアップすると、予想外の結果を招く可能性があります。

サブ組織の参照ポリシーを作成する

組織のポリシーに関する定義と決定を別の組織に委託することができます。この、ポリシーの作成と評価に関するポリシー委任は、参照ポリシーによって制御されます。これは、ルールと参照自体から構成されます。参照は、ルール内のリソースとして親組織を定義する必要があり、参照内の値として SubOrgReferral に組織名が含まれている必要があります。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。

デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。

2. ナビゲーションパネルで「アイデンティティ管理」を選択します。
3. 「表示」メニューから「ポリシー」を選択します。
4. 「新規」をクリックし、新しいポリシーを作成します。
ポリシーの作成ページがデータパネルに表示されます。
5. 「名前」には、**SubOrgReferral_Desktop** を入力します。「ポリシータイプ」で「参照」が選択されていることを確認します。「了解」をクリックします。
6. 「サービス」から「デスクトップ」を選択し、「次へ」をクリックします。
7. データパネルの「表示」メニューから「ルール」を選択し、「新規」をクリックします。「ポータルデスクトップ」が選択されていることを確認し、「次へ」をクリックします。
ルールの追加テンプレートがデータパネルに表示されます。
8. 「ルール名」に **DesktopRule** と入力し、「終了」をクリックします。
9. データパネルの「表示」メニューから「参照」を選択し、「新規」をクリックします。
「新規参照」テンプレートがデータパネルに表示されます。
10. 「名前」に **SubOrgReferral_Desktop** と入力します。
データパネルで「値」にサブ組織の名前が選択されていることを確認し、「作成」をクリックしてポリシーの設定を完了します。
11. データパネルで「保存」をクリックします。
データが保存されると、「ポリシープロパティが保存されました。」というメッセージが表示されます。

サブ組織に標準のポリシーを作成する

アクセス権の定義には標準のポリシーを使用します。標準のポリシーは、複数のルール、サブジェクト、条件から構成されます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ポリシーを作成する組織またはサブ組織に移動します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「表示」メニューから「ポリシー」を選択します。
その組織のポリシーが表示されます。
4. ナビゲーションパネルで「新規」を選択します。「新規ポリシー」ページがデータパネルに表示されます。
5. 「名前」に `SubOrgNormal_Desktop` と入力します。「ポリシータイプ」で「標準」が選択されていることを確認します。「作成」をクリックします。
6. データパネルの「表示」メニューから「ルール」を選択し、「新規」をクリックします。ルールの追加テンプレートがデータパネルに表示されます。
7. 「サービス」メニューから「ポータルデスクトップ」を選択し、「次へ」をクリックします。「ルール名」に `DesktopRule` と入力します。「NetMail 実行権限を所有」にチェックマークが付けられていることを確認します。
8. 「サービス」メニューから「ポータルデスクトップ」を選択し、「次へ」をクリックします。「NetMail 実行権限を所有」にチェックマークが付けられていることを確認します。
9. 「タイプ」メニューからサブジェクトのタイプを選択し、「次へ」をクリックしてサブジェクトの設定を完了します。
10. データパネルの「表示」メニューから「サブジェクト」を選択し、「新規」をクリックします。「サブジェクトを追加」ページがデータパネルに表示されます。
11. 「作成」をクリックして、ポリシーの設定を完了します。
データが保存されると、「ポリシープロパティが保存されました」というメッセージが表示されます。

ログインユーザーをポータルデスクトップ URL に正しくリダイレクトする

ログインに成功すると、組織内のユーザーはデフォルトでデスクトップサービスの属性、および値を受け取ります。これらの値はデスクトップサーブレットによって照会され、組織内の全ユーザーのポータルデスクトップコンテンツが決定されます。ユーザーがログインに成功したあと、Sun Java System Access Manager がポータルデスクトップサーブレットを自動的に呼び出すようにするには、デフォルトのリダイレクト URL をポータルデスクトップ URL に変更します。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

ポータルデスクトップ URL にリダイレクトされるように、特定の組織のデフォルトリダイレクトを設定するには、次の手順に従います。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ポータルデスクトップ URL を設定する組織を選択します。
3. 「表示」メニューから「サービス」を選びます。
4. ナビゲーションパネルの「コア」の隣にあるプロパティの矢印を選択します。
5. データパネルで、「デフォルト成功ログイン」という属性を探します。
6. 「デフォルト成功ログイン」の値をデスクトップサーブレットの URL に設定します。たとえば、/portal/dt はサンプルデスクトップの URL です。
7. 「保存」をクリックします。
8. デスクトップにログインして、デフォルトのポータルリダイレクト URL を確認します。

ログインユーザーをポータルデスクトップ URL にリダイレクトする (グローバル)

グローバル属性に適用される値は、Sun Java System Access Manager 設定に適用され、また新規に作成されたすべての組織に継承されます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

デフォルトリダイレクト URL をポータルデスクトップ URL にグローバルに設定するには、次の手順に従います。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「コア」の隣にあるプロパティの矢印を選択します。
4. データパネルで、「デフォルト成功ログイン」という属性を探します。
5. 「デフォルト成功ログイン」の値を /portal/dt などのポータルデスクトップサービスプレットの URL に設定します。
6. 「保存」をクリックします。

ポータルデスクトップサービス属性の値を修正する

サービス属性を修正することで、ポータルデスクトップサービスをカスタマイズできます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。

デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。

2. デスクトップ属性を修正する組織を選択します。
3. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティの矢印を選択します。

ディスプレイプロファイル XML を含む、ポータルデスクトップサービス属性の一覧がデータパネルに表示されます。

4. サービス属性値を修正します。
デスクトップ属性についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。
5. 完了したら、「保存」を選択します。

変更は、特定のサブ組織またはロールのユーザーだけに反映されます。

ポータルデスクトップサービス属性の値を修正する (グローバル)

場合によっては、今後、デスクトップサービスに追加するすべての組織に影響する、グローバルデスクトップサービスの属性値を修正する必要があります。

グローバル属性に適用される値は、Sun Java System Access Manager 設定に適用され、また今後構成されるすべての組織に継承されます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」をクリックします。
3. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティの矢印を選択します。
ディスプレイプロファイル XML を含むグローバルデスクトップサービス属性の一覧が、データパネルに表示されます。

4. サービス属性値を修正します。
デスクトップ属性についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。
5. 完了したら、「保存」を選択します。
この変更は、今後、デスクトップサービスに追加するすべての組織に影響します。

サンプルポータルデスクトップにアクセスする

1. Sun Java System Access Manager 管理コンソールからログアウトします。
2. 次の URL を使用してユーザーアカウント (amadmin ユーザーではない) でログインします。

`http://server:port/portal/dt`

ユーザーアカウントを作成する必要がある場合は、[89 ページの第 6 章「認証、ユーザー、およびサービスの管理」](#)を参照してください。

デスクトップログを確認する

ポータルデスクトップエラーはデバッグログファイルに記録されます。デフォルトでは、これらログファイルの位置は次のとおりです。

- `/var/opt/SUNWam/debug/desktop.debug`
- `/var/opt/SUNWam/debug/desktop.dpadmin.debug`

これらのログファイルのエラーを調べます。次に、例を示します。このエラーは認証されていないユーザーがポータルデスクトップを実行しようとしたことを示します。

```
06/20/2002 02:36:30:600 PM PDT:Thread[Thread-177,5,main]
ERROR:DesktopServlet.handleException()
com.sun.portal.desktop.DesktopException:DesktopServlet.doGetPost():no
privilige to execute desktop
    at
com.sun.portal.desktop.DesktopServlet.doGetPost(DesktopServlet.java:456)
    at
com.sun.portal.desktop.DesktopServlet.service(DesktopServlet.java:303)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
    at
com.sun.server.http.servlet.NSServletRunner.invokeServletService(NSServletRu
nner.java:897)
    at
com.sun.server.http.servlet.WebApplication.service(WebApplication.java:1065)
    at
com.sun.server.http.servlet.NSServletRunner.ServiceWebApp(NSServletRunner.ja
va:959)
```

ポートレットの管理

ポートレットの管理には、Sun Java™ System Access Manager 管理コンソールを使用します。管理コンソールには、ポートレットからポートレットチャンネルを作成するページ、およびポートレットチャンネルの設定を変更するページが用意されています。pdeploy は、ポートレット Web アプリケーションを Web コンテナに配備および配備取消しするのに使用できるコマンド行ツールです (コマンド行ユーティリティの管理についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照)。

注

少なくとも 1 つのセッションが有効なポートレットを含むポータルページにクライアント要求がアクセスするときは、そのポータルページのすべてのポートレットを 1 つのポートレットアプリケーションとしてパッケージ化することを強くお勧めします。パッケージしない場合、セッション作成の結果が 1 つに特定されなくなります。

ポートレットからチャンネルを作成する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールにログインし、組織を選択します。
2. ナビゲーションメニューの「表示」の下で「サービス」を選択します。
3. 「Portal Server 設定」から「ポータルデスクトップ」サービスを選択します。
4. 「チャンネルおよびコンテナを管理」リンクを選択します。
5. 「チャンネル」の下の「新規ポートレットチャンネル」ボタンを選択します。
ポートレットチャンネルを作成するためのページが表示されます。
6. 「チャンネルの追加」ページに次の情報を指定します。
 - チャンネル名。
 - チャンネル名は英数字 (a ~ z, A ~ Z, 0 ~ 9) で指定する必要があり、必須フィールドです。
 - ポートレット。
システムに配備されているポートレットだけが表示されます。
7. 「作成」ボタンを選択して、ポートレットチャンネルを作成します。

特定コンテナのポートレットからチャンネルを作成する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールにログインし、組織を選択します。
2. ナビゲーションメニューの「表示」の下で「サービス」を選択します。
3. 「Portal Server 設定」の下で「ポータルデスクトップ」サービスを選択します。

4. 「チャンネルおよびコンテナを管理」を選択します。
5. ポートレットチャンネルを作成するコンテナのリンクを選択します。
コンテナを管理するためのページが表示されます。
6. 「チャンネル」の下の「新規ポートレットチャンネル」ボタンを選択します。
ポートレットチャンネルを作成、追加するためのページが表示されます。
7. 「チャンネルの追加」ページに次の情報を指定します。
 - チャンネル名。
 - ポートレット(プルダウンリストから)。このリストには、システムに配備されているポートレットだけが含まれます。
8. 適切なラジオボタンを選択し、エンドユーザーがそのチャンネルを使用できるか、またはデスクトップ上で使用、表示できるかを指定します。
9. 「了解」ボタンを選択します。
「チャンネル」の下のチャンネルリスト、および「チャンネルおよびコンテナを管理」ページの「ポータルデスクトップで表示可能」の下にチャンネルが追加されます。

コンテナにポートレットチャンネルを追加する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールにログインし、組織を選択します。
2. ナビゲーションパネルの「表示」の下で「サービス」を選択します。
3. 「Portal Server 設定」から「ポータルデスクトップ」を選択します。
4. 「チャンネルおよびコンテナを管理」を選択します。
5. 新たに作成するポートレットチャンネルの追加先となるコンテナへのリンクを選択します。
コンテナを管理するためのページが表示されます。
6. 「使用可能」リストから追加するポートレットチャンネルを選択します。
「ポータルデスクトップで表示可能」の上に配置されている「追加」ボタンをクリックします。

これにより、選択しているコンテナで表示できるチャンネルのリストに、選択したポートレットチャンネルが追加されます。

7. 「チャンネル管理」の下の「保存」ボタンをクリックし、新しい設定を保存します。

ポートレットチャンネルの設定とプロパティを編集する

ポートレットの設定は、portlet.xml ファイルに定義されます。

```
<portlet-preferences>
    <preference>
        <name>foo</name>
        <value>apple</value>
    </preference>
    <preference>
        <name>bar</name>
        <value>orange</value>
        <value>grape</value>
        <read-only>true</read-only>
    </preference>
</portlet-preferences>
```

これは、次のディスプレイプロファイルにマッピングされます。

```
<Collection name="__Portlet__AdditionalPreferences"/>
    <Collection name="__Portlet__PreferenceProperties">
        <Collection name="default">
            <String name="foo" value="|apple|"/>
            <String name="bar" value="|orange|grape|"/>
        </Collection>
        <Collection name="isReadOnly">
            <Boolean name="foo" value="false"/>
            <Boolean name="bar" value="true"/>
        </Collection>
```

```

</Collection>
<String name="__Portlet_foo" value="|apple"/>
<String name="__Portlet_bar" value="|orange|grape"/>

```

実行時に追加される設定を保持するために、空のコレクション `__Portlet_AdditionalPreferences` が作成されます。コレクション `__Portlet_PreferenceProperties` には、`default` と `isReadOnly` という2つのコレクションが含まれます。`default` コレクションには、`portlet.xml` に定義されるデフォルト値が格納されます。`default` コレクションと似ている `isReadOnly` コレクションには、**Boolean** 型のプロパティを使用する設定の読み取り専用フラグが格納されます。

`portlet.xml` に含まれる各設定は、`default` コレクション内に対応する `String` プロパティを持ちます。このプロパティの名前は、設定名です。`String` プロパティの値は、`portlet.xml` に定義されているデフォルト値で、先頭に「|」という文字が付けられ、複数の値は「|」で区切られます。次に、各設定は設定の現在値を格納する `String` プロパティで表わされます。このプロパティの名前は、設定名で、先頭に `__Portlet_` という文字列が付けられます。プロパティの値は、現在のプロパティ値で、先頭に「|」という文字が付けられ、複数の値は「|」で区切られます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールにログインし、ドメインを選択します。
2. ナビゲーションパネルの「表示」の下で「サービス」を選択します。
3. 「Portal Server 設定」から「ポータルデスクトップ」を選択します。
4. 編集するポートレットチャンネルの「プロパティの編集」リンクを選択します。
5. 「チャンネルの編集」ページが表示されます。チャンネル編集ページには、ポートレットエンティティのポートレット設定が表示されます。
6. 設定を変更し、「保存」を選択して変更内容を保存します。
7. 設定のデフォルト値を変更するときは、編集する設定の「プロパティの編集」リンクをクリックします。プロパティは、「チャンネルの編集」ページで編集できません。

par ファイルの管理

par ユーティリティで、Portal Server 間でプロバイダまたはチャンネルを送信したり移動したりすることができます。par ユーティリティは、.par ファイルと呼ばれる特定のパッケージの仕組みを作成して、チャンネル、ポートレット、プロバイダをサーバー間で送受信します。.par ファイルは .jar ファイル形式の拡張形式で、ターゲットサーバーの Portal Server ディスプレイプロファイルに統合するための配備情報と XML ドキュメントが含まれる追加のマニフェスト情報を持っています。

par コマンドラインユーティリティを使用して par ファイルを作成、修正、配備します。export サブコマンドを使用すると par ファイルを作成、または修正できます。import サブコマンドを使用すると、Portal Server でプロバイダ、チャンネル、ポートレットをインポートまたは配備できます。describe サブコマンドを使用すると、par ファイルの内容の説明を参照できます。par コマンドの構文に関する詳細については、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

par ユーティリティを使用するには、エクスポートまたはインポートするファイルがある Portal Server に、superuser としてログインする必要があります。エクスポートする場合、チャンネル、ポートレット、またはプロバイダが要求するすべてのファイルを実際にエクスポートする必要があります。たとえば、チャンネルにはスタティックなコンテンツファイルを、プロバイダにはプロバイダが使用するすべてのクラスファイルをインクルードする必要があります。コマンドラインで par ファイルに含まれるすべてのデータを指定すると操作が煩雑になるので、データを示す行を持つ簡単なテキストファイルを作成し、par ユーティリティでこの「エクスポートファイル」を呼び出します。詳細は、[第 8 章「ポータルデスクトップサービスの管理」](#)を参照してください。

新規の par ファイルを作成する

新規の par ファイルを作成してから、次の手順でチャンネル、ポートレット、またはプロバイダをエクスポートします。

1. チャンネル、ポートレット、またはプロバイダのエクスポート元となる Portal Server にログインします。
2. スクリプトがインストールされているディレクトリに移動します。次のように指定します。

```
cd PortalServer-base/SUNWps/bin
```

3. コマンドラインで `par export` コマンドとサブコマンドを次の引数を含めて入力します。作成する `par` ファイル名、エクスポートするディスプレイプロファイルドキュメントに対応するディレクトリサーバー名、および任意の数の (少なくとも 1 つ) のエクスポートファイルや `from` 以下に指定した引数を指定します。たとえば、`o=sesta.com,o=isp` から `mychannel.par` ファイルへ `mychannel` チャンネルをエクスポートするには、次のコマンドを入力します。

```
./par export mychannel.par "o=sesta.com,o=isp" from:channel  
mychannel
```

詳しい構文情報については、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

既存の par ファイルを修正する

既存の `par` ファイルを修正して、次の手順でチャンネル、ポートレット、またはプロバイダをエクスポートします。

1. チャンネル、ポートレット、またはプロバイダのエクスポート元となる Portal Server にログインします。
2. スクリプトがインストールされているディレクトリに移動します。次のように指定します。

```
cd PortalServer-base/SUNWps/bin
```

3. コマンドラインで、`modify` オプションを使用して `par export` コマンドとサブコマンドを次の引数とともに入力します。修正する `par` ファイル名、エクスポートするディスプレイプロファイルドキュメントに対応するディレクトリサーバー名、および任意の数の (少なくとも 1 つ) のエクスポートファイル、または `from` 以下に指定した引数です。たとえば、`mychannel.par` ファイルを修正してスタティックコンテンツファイル `/mycontent.html` を含めるには、次のコマンドを入力します。

```
./par export --modify mychannel.par "dc=sesta,dc=com" "from= file  
/mycontent.html"
```

par ファイルを配備する

Portal Server へ par ファイルをインポートし、次の手順でプロバイダまたはチャンネルをシステムに配備します。

1. プロバイダまたはチャンネルに par ファイルをコピーして Portal Server へインポートし、そこにプロバイダまたはチャンネルを配備します。
1. チャンネル、ポートレット、またはプロバイダのインポート先となる Portal Server へログインします。
2. スクリプトがインストールされているディレクトリに移動します。次のように指定します。

```
cd PortalServer-base/SUNWps/bin
```

3. コマンドラインで par import コマンドとサブコマンドを次の引数を含めて入力します。インポートする par ファイル名、エクスポートするディスプレイプロファイルドキュメントに対応するディレクトリサーバー名です。たとえば、mychannel.par ファイルをインポートするには、次のコマンドを入力します。

```
./par import --auto myfile.par "do=sesta,dc=com"
```

Web Services for Remote Portlets (WSRP) サービスの管理

この章では、Sun Java™ System Portal Server の Web Services for Remote Portlets (WSRP) サービスの管理方法について説明します。

この章で説明する内容は次のとおりです。

- [WSRP 標準の概要](#)
- [WSRP プロデューサの管理](#)
- [WSRP コンシューマの管理](#)

WSRP 標準の概要

WSRP 1.0 は OASIS 標準の 1 つで、これによって、リモートアプリケーションおよびコンテンツのポータルへの統合が簡単にできます。WSRP 標準は、プレゼンテーション指向の対話式 Web サービスを、一般的な定義済みのインタフェースおよびプロトコルと共に定義して、ユーザー対話を処理し、プレゼンテーションを、ポータルによる仲介と集約、およびそれらのサービスを発行、検索、バインドするための規約に適した形にフラグメント化します。

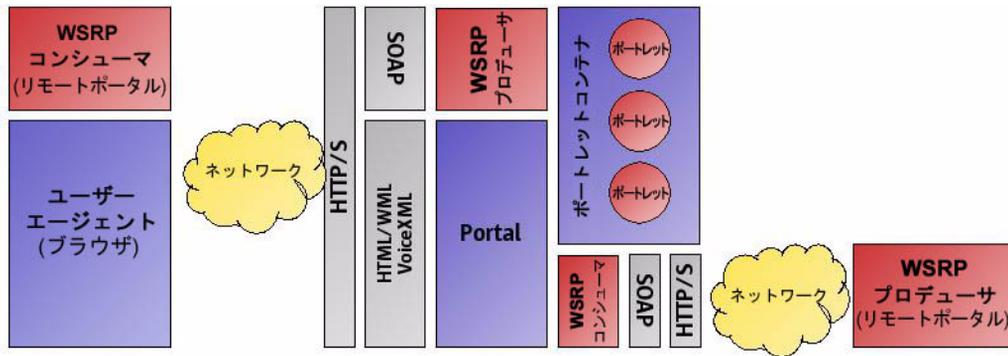
WSRP インタフェースは一般的で定義済みであるため、WSRP 標準を実装するすべての Web サービスは WSRP 準拠のすべてのポータルに接続できます。すなわち、あらゆる WSRP サービスを統合するには、サービスに依存しない単一のアダプタがポータル側にあれば十分です。したがって、WSRP は、コンテンツおよびアプリケーションプロバイダが、ポータルを実行している組織に対して複雑なプログラミング作業なく各自のサービスを提供するための手段となります。

詳細については、WSRP 1.0 標準を参照してください。

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp

Portal Server における WSRP 1.0 標準の実装には、WSRP のコンシューマとプロデューサの両方が含まれます。WSRP プロデューサの実装は、リモートの WSRP コンシューマによって使用される JSR 168 ポートレットの発行 (JSR 168 ポートレットの詳細は「[ポートレットの管理](#)」を参照) をサポートします。JSR 168 ポートレットは、ポータルサーバー上にローカルに配備されています。これらのポートレットは、WSRP プロデューサのインスタンスによって発行できます。もう 1 つのポータルサーバーは、WSRP コンシューマを介してこれらのリモートポートレットに登録できます。ローカルポートレットがポータルの基本機能の大部分を提供すると予想される一方で、リモートポートレットは、インストール作業、あるいはコンシューマ側のポータルサーバーでコードをローカルに実行する必要もなく、各種のリモートポートレットをバインドする可能性を提供します。

図 9-1 リモートポートレットの Web サービス



WSRP プロデューサの管理

ここでは、WSRP プロデューサを管理する作業について説明します。

- [WSRP プロデューサインスタンスを追加する](#)
- [WSRP プロデューサインスタンスを編集する](#)
- [WSRP コンシューマ登録を追加する](#)
- [WSRP コンシューマ登録を編集する](#)
- [すべての WSRP プロデューサを無効にする](#)

WSRP プロデューサインスタンスを追加する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 目的の組織またはサブ組織に移動して、「表示」メニューの「サービス」を選択します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「リモートポートレット (WSRP)」の「WSRP プロデューサ」の隣にあるプロパティの矢印を選択します。
データパネルに「WSRP プロデューサ」属性が表示されます。

注 要件やシステム構成に応じて、同じ組織またはサブ組織 (異なる組織またはサブ組織の場合もある) の下に複数の WSRP プロデューサを作成および有効化できます。この場合、WSRP プロデューサの名前が Portal Server システム全体で一意でなければならないという制約があります。

4. 「プロデューサ」の「新規」をクリックして、プロデューサインスタンスを追加します。
「基本パラメータの指定」ページが表示されます。
5. 次のプロパティの値を入力します。
 - プロデューサインスタンスの名前 (Portal Server 全体で一意でなければならない)
 - 登録が必要
「登録が必要」を選択した場合、すべての WSRP コンシューマは要求を出す前に、このプロデューサインスタンスに登録する必要があります。登録されていないコンシューマからの要求は拒否されます。
6. 「登録が必要」を選択した場合、「登録パラメータの指定」が表示されます。次のどちらかを指定します。

- in-band 登録がサポートされている (WSRP コンシューマがプロデューサインスタンスにオンラインで登録できる場合)。

in-band 登録によって、WSRP コンシューマのプログラムでの登録が可能になります。一方、out-of-band 登録では、登録ハンドルへのアクセス権の設定および交換のために、WSRP コンシューマ管理者と WSRP プロデューサ管理者の間で手動による連絡 (電子メール、電話など) が必要です。

- 登録時に WSRP コンシューマが提供する登録プロパティの説明。

WSRP コンシューマが登録時に提供する必要のある登録プロパティ。

注 登録検証は、登録検証インタフェースを実装する Java クラスです。必要に応じて、自分の登録検証クラスを作成し、コンシューマ登録の検証に必要なロジックを適用することができます。

7. 「次へ」をクリックします。
「ページの確認」のページが表示されます。
8. 提供した情報を確認して、「終了」をクリックします。

WSRP プロデューサインスタンスを編集する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 目的の組織またはサブ組織に移動して、「表示」メニューの「サービス」を選択します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「リモートポートレット (WSRP)」の「WSRP プロデューサ」の隣にあるプロパティの矢印を選択します。
データパネルに「WSRP プロデューサ」属性が表示されます。
4. 修正するプロデューサインスタンスの隣の「プロパティの編集」リンクを選択します。

5. 「一般プロパティ」タブを選択し、必要に応じてプロパティを修正します。
 - プロデューサインスタンスの状態：WSRP プロデューサがアクティブな場合は有効、プロデューサインスタンスがアクティブでない場合は無効です。

無効にしたプロデューサインスタンスは、WSRP コンシューマからの要求を受け付けません。デフォルトでは、1 つ以上のポートレットが発行されるまで WSRP プロデューサは無効になっています。
 - 登録が必要：WSRP コンシューマに登録が必要な場合はチェックマークを付けます。
 - in-band 登録：WSRP コンシューマがプロデューサインスタンスにプログラムで登録できる場合はサポートされます。WSRP コンシューマが WSRP プロデューサに手動 (out-of-band) でしか登録できない場合はサポートされません。

in-band 登録を使用すると、WSRP プロデューサは連絡や登録データの受け渡しをプログラム上で実行でき、WSRP コンシューマは WSRP プロデューサから登録ハンドルを受信できます。out-of-band 登録の場合、登録ハンドルへのアクセス権の設定および交換のために、WSRP コンシューマ管理者と WSRP プロデューサ管理者の間で手動による連絡 (電子メール、電話など) が必要です。
 - 登録検証クラス：登録の検証に使用される Java クラス。コンシューマがプロデューサに登録すると、プロデューサがサービスの説明に定義した登録プロパティごとに、コンシューマによって登録プロパティの値が渡されます。登録検証クラスは、コンシューマから送信された値が現在のプロデューサインスタンスで受け入れ可能かどうかをプロデューサが検証するために使用されます。

ヒント 登録検証は、登録検証インタフェースを実装する Java クラスです。必要に応じて、自分の登録検証クラスを作成し、コンシューマ登録の検証に必要なロジックを適用することができます。

6. 「保存」をクリックします。
7. 「ポートレット」タブを選択して、WSRP コンシューマから使用可能となる、配備されたポートレットの一覧を修正します。
8. 登録がサポートされている場合は、「登録プロパティ」タブを選択して、プロデューサインスタンスへの登録のためにすべての WSRP コンシューマが提供する必要のあるプロパティを追加または削除し、「保存」をクリックします。

WSRP コンシューマ登録を追加する

コンシューマ登録を行うと、WSRP コンシューマは WSRP プロデューサに対する自身の機能を説明できます。WSRP コンシューマは、out-of-band (電子メール、電話など) で追加されます。コンシューマ登録を追加するときに入力される情報は、登録ハンドルを与えられる WSRP コンシューマの機能と一致する必要があります。コンシューマ登録を行うと、WSRP プロデューサは、WSRP コンシューマが WSRP プロデューサ上に作成した成果物 (ポータル設定など) をスコープできます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 目的の組織またはサブ組織に移動して、「表示」メニューの「サービス」を選択します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「リモートポートレット (WSRP)」の「WSRP プロデューサ」の隣にあるプロパティの矢印を選択します。
データパネルに「WSRP プロデューサ」属性が表示されます。
4. 修正する WSRP プロデューサの隣の「プロパティの編集」リンクを選択します。
5. 「コンシューマ登録」タブを選択します。
6. 「コンシューマ登録」の「新規」をクリックします。
「新しいコンシューマ登録の作成」ページが表示されます。
7. 基本的なプロパティを必要に応じて指定します。
 - WSRP コンシューマの名前。
 - コンシューマ登録の状態 (コンシューマ登録がアクティブな場合は有効、アクティブでない場合は無効)。コンシューマ登録がアクティブでない場合、プロデューサは、無効なコンシューマ登録の登録ハンドルを含む要求を受け付けません。
 - コンシューマエージェント (アプリケーション名およびバージョンの ID)。
ID の形式は、*productName.majorVersion.minorVersion* です。たとえば、**Sun Java (tm) System Portal Server.6.3** のようになります。

- メソッド="get" (WSRP コンシューマが、メソッド="get" のフォームを含む HTML マークアップをサポートする形式でポータル URL を実装している場合はサポートされ、そうでない場合はサポートされない)。
8. 「次へ」をクリックします。

WSRP プロデューサが登録を必要とするように設定されている場合、「登録パラメータの指定」ページが表示されます。
 9. 登録プロパティの値を指定します。
 10. 「次へ」をクリックします。

「ページの確認」のページが表示されます。
 11. 提供した情報を確認して、「終了」をクリックします。

WSRP コンシューマ登録を編集する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。

デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 目的の組織またはサブ組織に移動して、「表示」メニューの「サービス」を選択します。

作成したすべての組織がナビゲーションパネルに表示されます。
3. 「リモートポートレット (WSRP)」の「WSRP プロデューサ」の隣にあるプロパティの矢印を選択します。

データパネルに「WSRP プロデューサ」属性が表示されます。
4. 修正する WSRP プロデューサの隣の「プロパティの編集」リンクを選択します。
5. 「コンシューマ登録」タブを選択します。
6. 修正するコンシューマ登録の隣の「プロパティの編集」リンクを選択します。

「コンシューマ登録の編集」ページが表示されます。
7. 必要に応じてプロパティを修正します。
 - WSRP コンシューマの名前。

- コンシューマ登録の状態 (コンシューマ登録がアクティブな場合は有効、アクティブでない場合は無効)。コンシューマ登録がアクティブでない場合、プロデューサは、無効なコンシューマ登録の登録ハンドルを含む要求を受け付けません。
- コンシューマエージェント (アプリケーション名およびバージョンの ID)。

ID の形式は、*productName.majorVersion.minorVersion* です。たとえば、**Sun Java (tm) System Portal Server.6.3** のようになります。

- メソッド = "get" (WSRP コンシューマが、メソッド = "get" のフォームを含む HTML マークアップをサポートする形式でポータル URL を実装している場合はサポートされ、そうでない場合はサポートされない)。
- コンシューマモード (WSRP コンシューマが管理しようとしているモードの配列。詳細については、WSRP 1.0 の仕様を参照)。
- コンシューマのウィンドウ状態 (WSRP コンシューマが管理しようとしているウィンドウ状態の配列。詳細については、WSRP 1.0 の仕様を参照)。
- コンシューマのユーザースコープ (WSRP コンシューマがユーザーコンテキストのために処理しようとしている値。詳細については、WSRP 1.0 の仕様を参照)。
- カスタムユーザープロファイルデータ (ユーザープロファイルの拡張子を指定する文字列の配列。詳細については、WSRP 1.0 の仕様を参照)。
- 登録プロパティ (登録プロパティのリスト。詳細については、WSRP 1.0 の仕様を参照)。

8. 「保存」をクリックします。

すべての WSRP プロデューサを無効にする

1. ロケーションパネルで「サービス設定」を選択します。
2. 「リモートポートレット (WSRP)」の「WSRP プロデューサ」の隣にあるプロパティの矢印を選択します。

データパネルに「WSRP プロデューサ」属性が表示されます。

3. 「グローバル」の「すべての WSRP プロデューサを無効」を選択します。
4. 「保存」をクリックします。

WSRP コンシューマの管理

ここでは、WSRP コンシューマを管理する作業について説明します。

- リモートポートレットチャンネルを作成する
- WSRP コンシューマの一般プロパティを編集する
- 設定済みの WSRP プロデューサを追加する
- 設定済みの WSRP プロデューサを編集する
- すべての WSRP コンシューマを無効にする
- 標準ユーザープロファイルのマッピングを編集する
- コンシューマ名を指定する

リモートポートレットチャンネルを作成する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 目的の組織またはサブ組織に移動して、「表示」メニューの「サービス」を選択します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. ポータルデスクトップサービスの隣にあるプロパティの矢印をクリックします。
データパネルにポータルデスクトップの属性が表示されます。
4. 「チャンネルおよびコンテナを管理」リンクをクリックします。
「チャンネル」ページが表示されます。ページの上部にコンテナのパスが表示されます。定義済みのチャンネルがリストに表示されます。
5. 「新規リモート (WSRP) チャンネル」をクリックしてリモートチャンネルを追加します。
「新規チャンネル」ページが表示されます。

6. 「プロデューサ」リストから「WSRP プロデューサ」を選択します。
7. 「ポートレット」リストからリモートポートレットを選択します。
8. 「チャンネル名」にリモートチャンネルの名前を入力します。
9. 「了解」をクリックします。

WSRP コンシューマの一般プロパティを編集する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 目的の組織またはサブ組織に移動して、「表示」メニューの「サービス」を選択します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「リモートポートレット (WSRP)」の「WSRP コンシューマ」の隣にあるプロパティの矢印を選択します。
データパネルに「WSRP コンシューマ」属性が表示されます。
4. 「一般」タブをクリックします。
5. 必要に応じてプロパティを修正します。
 - 名前: WSRP コンシューマは、登録作業中にコンシューマ名をプロデューサに送信します。
6. 「保存」をクリックします。

設定済みの WSRP プロデューサを追加する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 目的の組織またはサブ組織に移動して、「表示」メニューの「サービス」を選択します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「リモートポートレット (WSRP)」の「WSRP コンシューマ」の隣にあるプロパティの矢印を選択します。
データパネルに「WSRP コンシューマ」属性が表示されます。
4. 「設定済みのプロデューサ」タブをクリックします。
5. 「プロデューサ」の「新規」をクリックして、設定済みの WSRP プロデューサを追加します。
「基本パラメータの指定」ページが表示されます。
6. 新規に設定する WSRP プロデューサの名前と、目的の WSRP プロデューサの WSDL の URL を指定します。
7. 「次へ」をクリックします。
8. 目的の WSRP プロデューサが登録を必要として、in-band 登録をサポートしている場合は、WSRP コンシューマが次のどちらを実行するかを指定します。
 - 目的の WSRP プロデューサにオンラインでプログラムによって登録する
 - 目的の WSRP プロデューサから手動の方法で入手した登録ハンドルを提供する

in-band 登録を使用すると、WSRP プロデューサは連絡や登録データの受け渡しをプログラム上で実行でき、WSRP コンシューマは WSRP プロデューサから登録ハンドルを受信できます。out-of-band 登録の場合、登録ハンドルへのアクセス権の設定および交換のために、WSRP コンシューマ管理者と WSRP プロデューサ管理者の間で手動による連絡 (電子メール、電話など) が必要です。登録は必要ですが、in-band 登録が使用できない場合は、out-of-band で入手した登録ハンドルを入力する必要があります。
9. 「次へ」をクリックします。

10. 「登録プロパティ」に要求される情報を入力します。

WSRP プロデューサは、登録している WSRP コンシューマに、キーの値 (または、会社名、郵便番号など、プロデューサがサービス説明の中に示した登録プロパティ) を返すよう要求する可能性があります。プロデューサが登録プロパティを定義する場合、それらのプロパティはこのページに表示され、値が入力されています。

11. 「次へ」をクリックします。
「ページの確認」のページが表示されます。
12. 提供した情報を確認して、「終了」をクリックします。

注 追加した設定済みの WSRP プロデューサを編集して、ユーザーカテゴリとロールのマッピングを設定する必要がある場合があります。そうしないと、一部のポートレットが正しく動作しない可能性があります。

設定済みの WSRP プロデューサを編集する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 目的の組織またはサブ組織に移動して、「表示」メニューの「サービス」を選択します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「リモートポートレット (WSRP)」の「WSRP コンシューマ」の隣にあるプロパティの矢印を選択します。
データパネルに「WSRP プロデューサ」属性が表示されます。
4. 「設定済みのプロデューサ」タブをクリックします。
5. 修正する設定済みの WSRP プロデューサの隣の「プロパティの編集」リンクを選択します。
「基本プロパティの編集」ページが表示されます。

6. 必要に応じてプロパティを修正します。
 - WSRP プロデューサの名前
 - WSRP プロデューサの状態: WSRP プロデューサがアクティブな場合は有効、アクティブでない場合は無効です。

無効なプロデューサを対象とするリモートポートレットチャンネルは、「ポータルデスクトップ」上には表示されません。WSRP プロデューサを無効にする場合には注意が必要です。
 - サービスの説明: 「サービスの説明の更新」をクリックして、WSRP プロデューサの最新のサービスの説明を取得します。
7. 「保存」をクリックします。
8. 「ユーザーカテゴリとロールのマッピング」のリンクが表示された場合は、クリックします。

「ユーザーカテゴリとロールのマッピング」ページが表示されます。
9. WSRP ユーザーカテゴリが表示された場合、ローカルの Access Manager ロールに割り当てます。

注 すべての WSRP プロデューサが WSRP ユーザーカテゴリをサポートするわけではありません。

10. 「保存」をクリックします。
11. 「登録プロパティ」リンクが表示された場合は、クリックします。

「登録プロパティ」ページが表示されます。
12. 登録プロパティの値を必要に応じて修正します。
13. 「保存」をクリックします。

すべての WSRP コンシューマを無効にする

1. ロケーションパネルで「サービス設定」を選択します。
2. 「リモートポートレット (WSRP)」の「WSRP コンシューマ」の隣にあるプロパティの矢印を選択します。

データパネルに「WSRP コンシューマ」属性が表示されます。
3. 「グローバル」の「すべての WSRP コンシューマを無効」を選択します。
4. 「保存」をクリックします。

標準ユーザープロファイルのマッピングを編集する

Portal Server は、現在のユーザーの動作をパーソナライズする 1 つの方法として、エンドユーザー属性の標準セットを維持しています。

1. ロケーションパネルで「サービス設定」を選択します。
2. 「リモートポートレット (WSRP)」の「WSRP コンシューマ」の隣にあるプロパティの矢印を選択します。

データパネルに「WSRP コンシューマ」属性が表示されます。

3. 次の形式を使用して、「グローバル」の「標準ユーザープロファイルのマッピング」の値を修正します。

WSRP 定義のプロファイル |LDAP 属性

詳細については、WSRP 1.0 の仕様を参照してください。

4. 「保存」をクリックします。

コンシューマ名を指定する

WSRP コンシューマは、登録作業中にコンシューマ名をプロデューサに送信します。ここでコンシューマ名として指定された値は、組織またはサブ組織レベルで他の値がコンシューマ名として指定されない限り、デフォルトとして使用されます。

1. ロケーションパネルで「サービス設定」を選択します。
2. 「リモートポートレット (WSRP)」の「WSRP コンシューマ」の隣にあるプロパティの矢印を選択します。

データパネルに「WSRP コンシューマ」属性が表示されます。

3. 「コンシューマ名」に値を入力します。
4. 「保存」をクリックします。

ディスプレイプロファイルの管理

この章では、Sun Java™ System Portal Server のディスプレイプロファイルコンポーネントとそのコンポーネントの管理方法について説明します。

この章で説明する内容は次のとおりです。

- [ディスプレイプロファイルの概要](#)
- [ディスプレイプロファイルオブジェクトの集約](#)
- [ディスプレイプロファイルオブジェクトの検索](#)
- [ディスプレイプロファイルのプロパティ](#)
- [ディスプレイプロファイルのマージ動作](#)
- [ディスプレイプロファイルと Sun Java System Access Manager](#)
- [ディスプレイプロファイルの管理](#)

ディスプレイプロファイルの概要

ここでは Portal Server のディスプレイプロファイルコンポーネントについて説明します。

ディスプレイプロファイルは、次の 3 つの項目を定義してデスクトップの表示設定を作成します。

- **プロバイダ定義** : プロバイダの名前と Java™ クラスを指定します。プロバイダは、コンテンツの生成に使用されるテンプレートで、チャンネルに表示されます。詳細については、「[プロバイダオブジェクト](#)」を参照してください。
- **チャンネル定義** : プロバイダクラスのインスタンスの実行時設定を指定します。チャンネルはコンテンツの単位で、通常は (必ずしもそうとは限らないが) 行と列で編成されます。またチャンネル内にチャンネルを持つことができます。すなわちそれはコンテナチャンネルです。

- プロバイダとチャンネルのプロパティ定義**: プロバイダとチャンネルのプロパティに値を指定します。プロバイダに定義されたプロパティは、通常はそのプロバイダに関連するチャンネルの、デフォルトの値を指定します。チャンネルの表示設定には、タイトル、説明、チャンネル幅などのプロパティが含まれます。チャンネルに定義されるプロパティは、そのチャンネルに通常はデフォルトの値と異なる値を指定します。

コンテナプロパティは、コンテナ内に含まれるチャンネルの表示方法に関する表示定義を定義します。これには、コンテナのレイアウト (狭い - 広い、広い - 狭い、または狭い - 広い - 狭い)、コンテナ内のチャンネルのリスト、チャンネルの位置 (行と列の番号)、およびコンテナ内のチャンネルのウィンドウ状態 (最小または切り離し) が含まれます。

注	<p>ディスプレイプロファイルは、実際にはユーザーのデスクトップに表示される全体的なレイアウトまたは構造を定義しません。ディスプレイプロファイルはチャンネルのプロパティの値を提供するためにのみ存在します。ただし、ディスプレイプロファイルは、テーブルコンテナの列レイアウトやテーブルコンテナがチャンネルをテーブルの中に描画する方法のように、いくつかのチャンネルプレゼンテーションを間接的に制御します。</p> <p>ディスプレイプロファイルは、チャンネルのプロパティが配置を決定する場合のレイアウトを決定します。たとえば、サンプルポータルテーブルプロバイダ定義のディスプレイプロファイルには、次の文が指定されています。</p> <pre><Integer name="layout" value="1"/></pre> <p>これは「狭い - 広い」列を参照します。ただし、ディスプレイプロファイルの構造には実際のレイアウトに関する情報はありません。</p> <p>ディスプレイプロファイルは、XMLProvider が XML を解析する方法などは制御しません。その中にあるルール (XSL ファイル) の種類の定義があるだけです。</p>
----------	--

ポータルデスクトップは、Sun Java™ System Access Manager サービスのトップでディスプレイプロファイルデータ保管の仕組みを実装し、コンテンツプロバイダとチャンネルデータを保存します。さらに、チャンネルとプロバイダについてプロパティが設定されます。

ユーザーのディスプレイプロファイルは、コンテナ管理とチャンネルのプロパティが記述された一連の XML ドキュメントです (1 つのディスプレイプロファイルドキュメントが 1 つの XML ドキュメントに相当)。ディスプレイプロファイルドキュメントは、Sun Java System Access Manager サービス層の 1 つの属性としてその全体の中に格納されます。すなわち、ディスプレイプロファイルドキュメントは Sun Java™ System Directory Server のインスタンスに常駐する LDAP 属性です。

ディスプレイプロファイルプロパティ値を変更する場合、プロバイダはプロバイダ API (PAPI) を使用して値を取得および設定します。チャンネル値がディスプレイプロファイルに設定されると、PAPI 内部実装は Access Manager SDK を使用して Sun Java System Access Manager デスクトップサービス属性にディスプレイプロファイルドキュメントを設定します。

警告 可能な場合でも、Sun Java System Access Manager SDK を使ってディスプレイプロファイルを編集しないでください。

プロファイルと管理コンソールの表示

ディスプレイプロファイルと管理コンソールディスプレイプロファイルとその他のポータルデスクトップサービスのデータは、Sun Java System Access Manager 管理コンソールと dpadmin コマンドによって編集できます。ディスプレイプロファイルを編集する場合、プロバイダ、コンテナ、およびチャンネルを追加、修正、削除し、プロパティを編集します。「XML のアップロード」と「XML のダウンロード」リンクを使用して、ディスプレイプロファイルドキュメントをアップロードおよびダウンロードします。また、Sun Java System Access Manager 管理コンソールは、チャンネルとコンテナを追加し、既存のプロパティを編集するための「チャンネルおよびコンテナを管理」リンクを「ポータルデスクトップ」属性ページに表示します。チャンネルまたはコンテナが新規作成された場合、「チャンネルおよびコンテナを管理」リンクを使用してプロパティを定義します。また「チャンネルおよびコンテナを管理」リンクは、チャンネルとコンテナの追加、修正、削除にも使用できます。詳細については、「[ディスプレイプロファイルの管理](#)」を参照してください。

注 「チャンネルおよびコンテナを管理」リンクを使用した場合、ディスプレイプロファイルの一部にだけアクセスできるので、委任された管理者がこのリンクを使用すると想定されます。委任された管理者を設定する方法については、[第 7 章「管理の委任の設定](#)」を参照してください。

ディスプレイプロファイルドキュメントの構造

ここではディスプレイプロファイルドキュメントの全体的な構造について説明します。ディスプレイプロファイルドキュメントの基本となるデータ形式はXMLです。ディスプレイプロファイル DTD 構文については、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

ディスプレイプロファイル形式で、プロバイダとチャンネルオブジェクト、およびそのプロパティが定義され、デスクトップの表示設定が決定されます。ディスプレイプロファイルは Sun Java System Directory Server の isp レベル (または最上位ディレクトリノード)、組織レベル、ロールレベル、ユーザーレベルで保存されます。実行時にユーザーのディスプレイプロファイルは、ディレクトリツリー内のユーザー固有のプロファイルからディスプレイプロファイルドキュメントをすべて「マージ」したものであり、そのユーザーに関する特定のディスプレイプロファイルオブジェクトの値は、ディスプレイプロファイルの「マージ」動作により決定されます。

ディスプレイプロファイルオブジェクトは、オブジェクトを定義する XML タグに直接マッピングされます。たとえば、<Channel name></Channel> XML タグはチャンネルオブジェクトを定義します。

一般に、ディスプレイプロファイルのドキュメント構造は、次のようになります。

```
<DisplayProfile>
  <Properties>...global properties...</Properties>
  <Channels>...channel definitions...</Channels>
  <Providers>...provider definitions...</Providers>
</DisplayProfile>
```

<Properties>、<Channels>、および <Providers> はグループ化を実行するためのメカニズムです。これらのメカニズムは XML ディスプレイプロファイルドキュメントを詳細に構造化し、オブジェクトなどは各「バッグ」内に格納されます。「バッグ」については、「[ディスプレイプロファイルオブジェクトの集約](#)」を参照してください。

次の節からはディスプレイプロファイルオブジェクトをより詳細に説明します。

ディスプレイプロファイルルートオブジェクト

ディスプレイプロファイルの root コンテナオブジェクトでは、デスクトップサーブレットはコンテナプロバイダとして機能し、プロバイダのハンドルの取得などを行います。チャンネルと関連付けられる実際のプロバイダクラスは存在しません。このチャンネルは他のディスプレイプロファイルオブジェクトから参照されることはありません。

ディスプレイプロファイルルートオブジェクトのXML 構文

```
<Container name="_desktopRoot" provider="none">
  <Properties/>
  <Available />
  <Selected />
  <Channels/>
</Container>
```

プロバイダオブジェクト

プロバイダオブジェクトは、チャンネルレンダリングの実行時に実行されるソフトウェアエンティティです(したがって、チャンネルは実行時にはプロバイダのインスタンスとなる)。<Provider>ディスプレイプロファイル定義は、ディスプレイプロファイルチャンネルを定義するためのテンプレートです。プロバイダ Java オブジェクトのクラス名とすべての必須プロパティのデフォルト値を設定します。

<Provider>ディスプレイプロファイル定義には、ディスプレイプロファイルのクライアントが Java™ クラス名などの provider オブジェクトを構築するために必要な情報が収められています。

<Provider>ディスプレイプロファイル定義は、このプロバイダを示すすべてのチャンネルに対して、デフォルトのプロパティ値を設定します。チャンネル固有のプロパティが必要になるのは、プロバイダのデフォルトを上書きする必要がある場合のみです。プロバイダディスプレイプロファイルオブジェクトには、プロバイダ Java オブジェクトで使用されるすべてのプロパティに対するデフォルトの値を格納する必要があります。たとえば、プロバイダ Java コードには次の値が格納されます。

```
getStringProperty("color")
```

チャンネルオブジェクト

channel オブジェクトは単一の表示要素を表します。channel オブジェクトに含まれるオブジェクトは、チャンネルのプロパティと考えることができます。<Channel name>定義には、プロバイダへのシンボル参照が含まれます。また、チャンネル固有のプロパティを定義して、プロバイダ定義で定義されるデフォルトの値を上書きすることができます。チャンネル名は、ディスプレイプロファイルドキュメント内の同じチャンネルレベルでは一意である必要がありますが、チャンネルレベルが異なれば、同じ名前を定義できます。

チャンネルオブジェクトのXML 構文の例

```
<Channel name="SampleXML" provider="XMLProvider">

  <Properties >
    <String name="refreshTime" value="600" advanced="true"/>
    <String name="title" value="XML Test Channel"/>
    <String name="description" value="This is a test of the XML Provider system"/>
    <String name="url"
value="file:///etc/opt/SUNWps/desktop/default/SampleXML/getQuotes.xml"/>
    <String name="xslFileName"
value="/etc/opt/SUNWps/desktop/default/SampleXML/html_stockquote.xsl"/>
  </Properties>

</Channel>
```

コンテナオブジェクト

container オブジェクトは channel オブジェクトに似ていますが、container オブジェクトはコンテンツを生成しません。すなわち、コンテナは他のチャンネルから内容を取得するチャンネルです。container オブジェクトを使用すると、チャンネルリストの使用と選択が可能になり、子チャンネル定義を格納することができます。子チャンネルは、一般にページ上で他のチャンネルと集約されて、独自の内容を生成します。コンテナチャンネルは、主に、1つまたは複数の子チャンネルの内容を集約してその内容を生成します。

コンテナオブジェクトのXML 構文の例

```
<Container
name="TemplateTableContainer"provider="TemplateTableContainerProvider">
  <Properties>
    <String name="title" value="Template Based Table Container"/>
    <String name="description"
      value="This is the channel for the front provider"/>
    <Collection name="channelsColumn" advanced="true">
      <String name="SampleJSP" value="2"/>
      <String name="SampleXML" value="2"/>
      <String name="Notes" value="2"/>
    </Collection>
    <Collection name="channelsRow" advanced="true">
      <String name="MailCheck" value="3"/>
      <String name="SampleRSS" value="2"/>
      <String name="SampleXML" value="2"/>
      <String name="App" value="5"/>
      <String name="SampleSimpleWebService" value="6"/>
      <String name="Bookmark" value="4"/>
      <String name="Notes" value="3"/>
    </Collection>
    <Collection name="channelsIsRemovable">
      <Boolean name="UserInfo" value="false"/>
    </Collection>
  </Properties>
  <Available>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    <Reference value="SampleRSS"/>
    <Reference value="SampleJSP"/>
    <Reference value="SampleXML"/>
    <Reference value="App"/>
    <Reference value="SampleSimpleWebService"/>
    <Reference value="Bookmark"/>
    <Reference value="Notes"/>
  </Available>
```

```

<Selected>
  <Reference value="UserInfo"/>
  <Reference value="MailCheck"/>
  <Reference value="SampleRSS"/>
  <Reference value="SampleJSP"/>
  <Reference value="SampleXML"/>
  <Reference value="App"/>
  <Reference value="SampleSimpleWebService"/>
  <Reference value="Bookmark"/>
  <Reference value="Notes"/>
</Selected>

<Channels>
</Channels>

</Container>

```

ディスプレイプロファイルオブジェクトの集約

root、provider、および channel オブジェクトには、プロパティを関連付けることができます。ディスプレイプロファイルは、プロパティ「バッグ(入れ物)」内のプロパティをグループ化します。バッグ(入れ物)という用語は、ここにはプロパティを保持するだけである、ということの意味するために使用します。プロパティにプロパティバッグが関連付けられることはありません。プロパティの定義については、『Sun Java System Portal Server 6 2004Q2 Desktop Customization Guide』を参照してください。

チャンネル、プロバイダ、およびルートレベルのプロパティバッグは、異なるセマンティクスを持ちます。グローバルプロパティは、すべてのチャンネルで共有されます。グローバルプロパティとして定義されるプロパティは、ここではどのチャンネルからでもアクセスできます。テーマはグローバルプロパティの一例です。テーマデータはグローバルに定義されるため、すべてのチャンネル間で共有されます。

プロバイダで定義されるプロパティは、そのプロバイダに基づくチャンネルのデフォルトです。プロパティがチャンネルで定義されていない場合、デフォルトが使用されます。これはプロバイダがプロバイダ Java オブジェクトで使用されるプロパティをすべて定義する必要があることを示しています。この場合、Java コードの内容は次のようになります。

```
String f = getStringProperty("color");
```

ディスプレイプロファイル内の対応する <Provider name> 定義では、次のように定義する必要があります。

```
<String name="color" ... />
```

注 グローバルプロパティを、すべてのチャンネルのデフォルトとして使用しないでください。ディスプレイプロファイルプロバイダ定義は、プロバイダ定義を使用する provider オブジェクトが使用するプロパティインタフェースを定義します。

チャンネルプロパティは、プロバイダ定義のデフォルトを上書きしてチャンネルをカスタマイズします。たとえば、URLScaperProvider は url プロパティを定義します。デフォルトは、通常はチャンネルによって値が上書きされるため意味を持ちません。

ディスプレイプロファイルオブジェクトの検索

実行時に、システムは直接プロバイダにプロパティを照会しません。この要求は常にチャンネルに対して行われます。Java provider オブジェクトがプロパティを要求する場合、プロパティが見つかるか、格納階層の最上位に到達するまで次の順序でディスプレイプロファイルを検索します。

1. チャンネルのプロパティ
2. チャンネルのプロバイダのプロパティ
3. チャンネルの親のプロパティ
4. チャンネルの親のプロバイダのプロパティ
5. チャンネルの親のプロパティ (など)
6. ディスプレイプロファイル root 定義に定義されるグローバルプロパティバッグ

このようにチャンネルがプロパティの名前を照会すると、上記のすべてに該当するプロパティが取得されます。

provider オブジェクトに存在するプロパティは、チャンネルに対してデフォルト値の動作を持つように意図されています。たとえば、プロパティ title を定義するプロバイダ xml について、プロバイダ xml から導出されるすべてのチャンネルは title プロパティを継承します。チャンネルがこのプロパティを上書きする場合、その独自のプロパティ内で値を設定できます。

ディスプレイプロファイルのプロパティ

ここでは、ディスプレイプロファイルプロパティとそれらを指定する方法について説明します。

ディスプレイプロファイルのプロパティのタイプ

ディスプレイプロファイルプロパティのタイプは次のとおりです。

- **Boolean:** ブール値を表す不可分なオブジェクト。
例：

```
<Boolean name="isEditable" value="false"/>
```
- **Collection:** リストまたはハッシュテーブルのいずれかを表すオブジェクト。コレクションは、別のプロパティを格納するためのプロパティのタイプ、つまり名前付きバッグです。
例：

```
<Collection name="channelsRow">  
    <String name="MailCheck" value="4"/>  
    <String name="App" value="5"/>  
</Collection>
```
- **Integer:** 整数値を表す不可分なオブジェクト。
例：

```
<Integer name="numberOfHeadlines" value="7"/>
```
- **String:** 文字列の値を表す不可分なオブジェクト。
例：

```
<String name="title" value="Table Container Channel 1"/>
```
- **Reference:** チャンネル定義 (すなわち、コンテナの選択された使用可能なチャンネルリスト内のチャンネル名) を示すポインタを表すオブジェクト。参照は、設計ツールに便利な名前のない文字列で、文字列を他のものと区別することができます。
例：

```
<Reference value="UserInfo"/>
```

不可分なプロパティ値は、本文コンテンツとしても指定できます。

例：

```
<String name="foo">bar</String>  
<Integer name="aNumber">1</Integer>  
<Boolean name="flag">>false</Boolean>
```

ドキュメントタイプの定義要素の属性

ポータルデスクトップ DTD は、ディスプレイプロファイルとそのプロパティの使用を制御できる要素属性を定義します。205 ページの表 10-1 は、ドキュメントタイプの定義要素の属性を示しています。この表には、最初の列に属性、2 番目の列に簡単な説明、3 番目の列に例が示されます。

表 10-1 ディスプレイプロファイルの属性

属性	定義	例
advanced	<p>true に設定した場合、Sun Java System Access Management 管理コンソールの「チャンネルおよびコンテナを管理」リンクでディスプレイプロファイルプロパティを非表示にします。ただし、「XML の直接編集」リンクまたは「XML のダウンロード」リンクを使用する場合は、プロパティは非表示になりません。</p> <p>advanced 属性は、値に true または false をとることができるブール型の属性です。デフォルト値は false です。</p>	<pre><String name="refreshTime" value="0" advanced="true"/></pre>
lock	<p>優先順位の低いドキュメントによって、優先順位の高いドキュメントがマージ操作でディスプレイプロファイルの特定の機能を変更できないようにします。ディスプレイプロファイルオブジェクトがロックされると、優先順位の低いドキュメントによるマージ操作の影響を受けなくなります。</p> <p>lock 属性は、値に true または false をとることができるブール型の属性です。デフォルト値は false です。</p>	<pre><Selected merge="fuse"> ... <Reference value="EmployeeNews" lock="true"/> ... </Selected></pre>
merge	<p>異なる LDAP ノード (base DN、DN と role DN) のディスプレイプロファイルドキュメントをマージして単一の表現 (すなわちポータルデスクトップ) を形成する場合に、プロパティの結合方法を制御します。</p> <p>許可される値は replace、remove、および fuse です。デフォルト値は fuse です。</p> <p>不可分なプロパティ (boolean int、stringv ref) では fuse は無効です。</p>	<p>replace、remove、fuse の例については、「ディスプレイプロファイルのマージタイプ」を参照してください。</p>

表 10-1 ディスプレイプロファイルの属性 (続き)

属性	定義	例
priority	<p>ディスプレイプロファイルドキュメントの優先順位を設定します。ディスプレイプロファイルドキュメントは、低い優先順位から高い優先順位の順にマージされます。低い数値は低い優先順位を表します。たとえば、1は2よりも優先順位が低くなります。</p> <p>マージ動作により、優先順位が高いドキュメントの値は、優先順位が低いドキュメントの値に優先して適用されます(ただし、優先順位の低いドキュメントがオブジェクトのマージをロックしている場合を除く)。</p> <p>許可される値は、整数と、キーワード <code>user</code> です。 <code>user</code> は最高の優先順位であり、ユーザーレベルのディスプレイプロファイルドキュメントだけに設定するべきです。</p>	<pre><DisplayProfile version="1.0" priority="10"></pre>
propagate	<p>プロパティがローカル以外に読み出され、ローカルに設定される場合の、プロパティの処理方法を制御します。ブール型、コレクション、整数、文字列、および参照を含むすべてのディスプレイプロファイルプロパティは、<code>propagate</code> 属性に設定できます。</p> <p><code>propagate</code> 属性は、値に <code>true</code> または <code>false</code> をとることができるブール型の属性です。デフォルト値は <code>true</code> です。</p>	<pre><String name="color" value="blue" propagate="false"/></pre>

ディスプレイプロファイル XML では、属性のデフォルト値を変更しない限り、次の属性は XML ファイルにリストされず、管理コンソールにも表示されません。

```
<advanced="false" lock="false" merge="fuse" propagate="true">
```

デフォルト値がリセットされると、デフォルト値が変更された属性だけが XML ファイルにインクルードされ、管理コンソールに表示されます。デフォルトプロパティはプロバイダから継承されます。デフォルトプロパティが編集されている場合は、カスタマイズされたプロパティとして表示されます。

ディスプレイプロファイルのプロパティの指定

ディスプレイプロファイルプロパティを指定する場合は、プロパティを「ネスト」する方法、コレクションに名前のないプロパティを使用する方法、条件付きプロパティを使用する方法、およびプロパティを伝達する方法を考慮する必要があります。

入れ子のプロパティ

ディスプレイプロファイルには任意の深さまで入れ子にしたプロパティ (プロパティ内のプロパティ) を格納することができます。これによって、文字列のコレクションのコレクションのコレクションや、文字列とコレクションのコレクションなどを作成することができます。たとえば、次に示すのはコレクションのコレクションです。

```
<Collection name="people">
  <Collection name="john">
    <Integer name="age" value="31"/>
    <String name="eyes" value="hazel"/>
  </Collection>
  <Collection name="bob">
    <Integer name="age" value="35"/>
    <String name="eyes" value="blue"/>
  </Collection>
  ... etc ...
</Collection>
```

名前を付けられていないプロパティ

次のような不可分のプロパティのタイプ (ブール型、整数、文字列) は名前が付けられていない場合があります。

```
<String value="apple"/>
```

これは次の値に等しくなります。

```
<String name="apple" value="apple"/>
```

つまり、不可分なプロパティが名前を持たない場合、名前はプロパティの文字列値となります。

テーブルではなくコレクションを使用して順序付けられたセット (リストに近い) を表すことができるので、実用上はコレクション内だけで便利に使用できます。たとえば、次のコレクションは郵便番号のリストを表します。

```
<Collection name="zipcodes">
  <Integer value="95112"/>
  <Integer value="95054"/>
  <Integer value="98036"/>
</Collection>
```

名前の付けられていないプロパティを使用する場合、テーブル (*name=value*) またはリストをコレクションで表すことができる点が重要です。

注 名前の付けられていないプロパティを、同じコレクション内の別の名前のないプロパティと同じ値で作成しないでください。プロパティは作成されますが、名前が重複しているため、プロバイダは値にアクセスすることができません。

また、**Portal Server** は同じ名前と同じ値を持つプロパティを名前のないブール型プロパティと同等に処理するため、同じコレクション内で名前が重複するプロパティを間違えて作成してしまう場合があります。この場合も、最終的に他の同じ名前を持つプロパティにアクセスできなくなります。

条件付きプロパティ

これは、条件付きプロパティを取得するための汎用処理を提供します。最も一般的な条件は、`locale` と `client` ですが、任意の条件にプロパティを定義することができます。詳細は、『*Sun Java System Portal Server 6 2004Q2 Desktop Customization Guide*』を参照してください。

たとえば、ロケールフィルタの実装は、次のように定義します。

```
public class LocalePropertiesFilter extends PropertiesFilter {
    public LocaleProperties() {
        super();
    }
    String getCondition()
    return "locale";
    }
    public boolean match(ProviderContext pc, String condition,
String value) {
    return condition.toLowerCase().equals("locale")    &&
        getValue().equals(value);
    }
}
```

条件付きプロパティの検索には、1つまたは複数のプロパティフィルタが関連します。フィルタリスト内のフィルタが必要な場合は、条件付き検索全体とそのフィルタが一致すれば検索は正常に行われます。フィルタを必要としない場合は、一致しなくても全体的な検索は失敗しません。

必須以外のフィルタの連鎖を使用して、具体性が低いフィルタ検索を段階的に実装することで、Java リソースバンドル検索と似た操作を実行できます。たとえば、ロケール検索の後に日付検索を行う場合は、オプションフィルタが便利です。{locale=en, locale=US, date=03/03/2003} というフィルタがある場合、フィルタの仕様と完全に一致するわけではありませんが、修飾子 {locale=en; date=03/03/2003} でプロパティと一致させることができます。これは、ロケールフィルタをオプションとして設定することで行われます。

管理コンソールでは、条件付きプロパティは condition-value として表示され、コレクションと同様に編集できます。条件付きプロパティは、入れ子にすることができ、チャンネルまたは別の条件付きプロパティに追加することができます。新しい条件付きプロパティを追加するときは、「新規プロパティ」ページを使用します。

<ConditionalProperties> タグ

フィルタリング条件を定義するときは、<ConditionalProperties> タグを使用する必要があります。このタグには、次の必須属性が含まれます。

- condition: フィルタに適用する条件を指定する
- value: フィルタに適用する値を指定する

ディスプレイプロファイルでは、[210 ページのコード例 10-1](#) で説明する方法で <ConditionalProperties> タグを定義できます。

コード例 10-1 <ConditionalProperties> タグの使用例

```

<Properties>
  <String name="foo" value="bar">
  <ConditionalProperties condition="locale" value="de">
    <String name="foo" value="german bar">
    <String name="baz" value="a german baz value">
  </ConditionalProperties>
  <ConditionalProperties condition="client" value="nokia">
    <ConditionalProperties condition="locale" value="de">
      <String name="foo" value="nokia german bar">
    </ConditionalProperties>
  </ConditionalProperties>
</Properties>

```

ディスプレイプロファイルのプロパティの伝達

ブール型、コレクション、整数、文字列、および参照を含むすべてのディスプレイプロファイルプロパティには、propagate 属性のマークを付けることができます。propagate 属性は、値に true または false をとることができるブール型の属性です (デフォルトは true)。propagate 属性は、プロパティがローカル以外に読み出され、ローカルに設定される場合の、プロパティの処理方法を制御します。

たとえば、チャンネルのプロパティセットは、次のすべてに該当するプロパティから構成されます。

- チャンネルのプロパティ (<Properties>) バッグにローカルに存在するプロパティのセット
- チャンネルのプロバイダ (チャンネルで provider 属性によって指定される) にローカルに存在するプロパティのセット
- チャンネルの各祖先チャンネル (チャンネルの親、チャンネルの親の親、など) にローカルに存在するプロパティのセット
- チャンネルの各祖先チャンネルプロバイダ (チャンネルの親プロバイダ、チャンネルの親の親プロバイダ、など) にローカルに存在するプロパティのセット
- ディスプレイプロファイル root オブジェクトの下に存在するグローバルプロパティのセット

チャンネルがプロパティ値を要求すると、値はこれらの「リモート」の場所のいずれかから読み出されます。プロパティ値が設定されている場合、プロパティ値を格納する場所は 2 つのオプションから選択できます。

1. チャンネルのプロパティバッグ
2. リモートロケーション

propagate 属性は場所を制御します。propagate 属性を true に設定すると、プロパティはプロパティを設定するオブジェクトのローカルに保存されます (ほとんどの場合はチャンネル)。propagate 属性を false に設定すると、プロパティは所定の位置 (読み出された場所) に設定されます。すなわち、false に設定された場合、既存の値は変更されますが、true に設定された場合、新規プロパティがローカルに作成されて保存されます (すでにローカルになっている場合を除く)。

次のディスプレイプロファイル XML コード例を考えてみます。

```
<DisplayProfile>
  <Properties>
    <String name="color" value="blue"/>
  </Properties>
  ...
  <Channel name="testchannel" provider="..."/>
  <Properties/>
</Channel>
...
</DisplayProfile>
```

プロパティ color はグローバルプロパティバッグに存在します。propagate は設定されていないため (またデフォルトは true であるため)、チャンネル testchannel でプロパティ color が設定されると、次のような結果になります。

```
<DisplayProfile>
  <Properties>
    <String name="color" value="blue"/>
  </Properties>
  ...
  <Channel name="testchannel" provider="..."/>
  <Properties/>
    <String name="color" value="new value"/>
  </Channel>
  ...
</DisplayProfile>
```

プロパティは、これを設定したローカルオブジェクト (チャンネル) に伝達されます。一方、グローバルプロパティバッグなどで propagate が false に設定されていると、次のようになります。

```
<String name="color" value="blue" propagate="false"/>
```

チャンネル testchannel の設定プロパティ color の結果は次のようになります。

```
<DisplayProfile>
  <Properties propagate="false">
    <String name="color" value="new value" />
  </Properties>
  ...
  <Channel name="testchannel" provider="..." />
  <Properties />
</Channel>
...
</DisplayProfile>
```

個々のプロパティの他に、プロパティバッグにも次のように propagate 属性を指定できます。

```
<Properties propagate="false">
  ...
</Properties>
```

プロパティが propagate=false と見なされる場合、以下は真である必要があります。

- プロパティの propagate 属性が false であるか、プロパティのプロパティバッグの propagate 属性が false である。
- 上記の文は、プロパティのすべてのマージ結果において真である必要があります。

その他の場合は、propagate は true と見なされます。

最上位のプロパティにのみ propagate 属性を表記することができます。ディスプレイプロファイル DTD ではこのような表記が可能ですが、ディスプレイプロファイルコードはこれを見捨てます。最上位のプロパティは、プロパティバッグ内に直接定義されます。

ディスプレイプロファイルドキュメントの優先順位

実行時に、ユーザーがログインすると、システムはユーザーのディスプレイプロファイルドキュメントセットを構成するドキュメントセットを決定します。ディスプレイプロファイルにおけるデスクトップの内部実装 (ディスプレイプロファイルを解釈する部分) では、ユーザーが所属する LDAP ノードすべてを検索して、このドキュメントセットを決定します。これには、組織 DN (o=sesta.com)、サブ組織、ロール DN (cn=Role1,o=sesta.com)、uid (uid=jtb,ou=People,cn=Role1,o=sesta.com)、およびグローバルディスプレイプロファイルが含まれます。これらの LDAP ノードとグローバルディスプレイプロファイルのそれぞれのディスプレイプロファイルドキュメントが読み込まれ (ドキュメントが存在する場合)、すべてのドキュメントが1つにまとめられます。システムはドキュメントの優先順位に従って、このセットをソートします。低い数値は低い優先順位を表します。たとえば、1は2よりも優先順位が低くなります。ドキュメントは低い数値から高い数値の順にソートされます。このプロセスについての詳細は、「[マージプロセスの仕組み](#)」を参照してください。

ユーザーレベルドキュメント (uid=jtb,ou=People,...) はベースドキュメントと呼ばれる特殊な例です。ベースドキュメントは優先順位が無最大であると考えてください。したがって常に最高位の数値 (および最高の優先順位) になります。すべてのマージ結果は、ソート順でベースドキュメントと関連付けられ、ユーザードキュメントの優先順位の設定は常に最高位になります。<DisplayProfile> タグで使用される priority 属性は特殊なキーワード user を使用して、現在のディスプレイプロファイルがユーザーレベルのディスプレイプロファイルであることを示します。

マージは、もっとも優先順位の低いドキュメント (もっとも低い数値) から開始され、ユーザー (ベース) ドキュメントに達するまで、優先順位の数値を増加させながら処理されます。

ディスプレイプロファイルドキュメントの優先順位が実際に意味するのは、重要なのは優先順位の数値であることです。たとえば、組織レベルのドキュメントの優先順位をロールレベルのドキュメントよりも高くできますが、必ずしもそのようにする必要はありません。優先順位の設定は、自分のサイトでこのようなドキュメントをどのような優先順位に設定する必要があるかによります。

ディスプレイプロファイルドキュメントの優先順位は、XML ファイルに <DisplayProfile priority=syntax> タグを使用して指定します。Sun Java System Access Manager 管理コンソールまたは dpadmin コマンドを使用してディスプレイプロファイルを読み、直接ディスプレイプロファイル XML を編集することで優先順位を変更します。コマンド行ユーティリティの管理についての詳細および dpadmin コマンドの詳細については、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

注 2つのディスプレイプロファイルドキュメントに同じ優先順位を割り当てないでください。デスクトップが適切に表示されなくなります。ただし、製品はドキュメントの優先順位の重複をチェックしません。

ドキュメントの優先順位の例 1

次の例では2つのディスプレイプロファイルを使用しています。1つは組織の acme、もう1つは uid の bill に使用します。Bill (uid=bill) がデスクトップにログインすると、ブックマークチャンネル「Bill's Bookmarks」が次の3つのブックマーク(次の順序で)と共に表示されます。

- ACME
- Amazon
- EBay

```
ディスプレイプロファイル @ o=acme.com
<DisplayProfile version="1.0" priority="10">
...
  <Channel name="Bookmark" provider="BookmarkProvider" merge="fuse">
    <Properties>
      <String name="title" value="My Bookmarks" merge="replace" lock="false"
propagate="true"/>
      <String name="refreshTime" value="600" merge="replace" lock="false"
propagate="true"/>
      <Collection name="targets" merge="fuse" lock="false" propagate="true">
        <String value="ACME home page|http://www.acme.com" merge="replace" lock="false"
propagate="true"/>
      </Collection>
    </Properties>
  </Channel>
...
</DisplayProfile>
```

```
dp @ uid=bill,ou=people,o=acme.com
<DisplayProfile version="1.0" priority="1">
  ...
  <Channel name="Bookmark" provider="BookmarkProvider" merge="fuse">
    <Properties>
      <String name="title" value="Bill's Bookmarks" merge="replace" lock="false"
propagate="true"/>
      <Collection name="targets" merge="fuse" lock="false" propagate="true">
        <String value="Amazon|http://www.amazon.com" merge="replace" lock="false"
propagate="true"/>
        <String value="EBay|http://www.ebay.com" merge="replace" lock="false"
propagate="true"/>
      </Collection>
    </Properties>
  </Channel>
  ...
</DisplayProfile>
```

ドキュメントの優先順位の例 2

次の例では、グローバルディスプレイプロファイル、組織 acme のディスプレイプロファイル、およびロール hradin のディスプレイプロファイルという 3 つのディスプレイプロファイルを使用します。hradin ロールに割り当てられたユーザーがデスクトップにログインすると、TemplateTableContainer が次のチャンネルとともに表示されます。

- UserInfo
- MailCheck
- SampleSimpleWebService

```
dp @ global:
<DisplayProfile version="1.0" priority="0">
  ...
  <Container name="TemplateTableContainer" provider="TemplateTableContainerProvider"
merge="fuse">
  <Properties>
    ...
  </Properties>
  <Available>
    ...
  </Available>
  <Selected merge="fuse" lock="false">
    <Reference value="UserInfo"/>
  </Selected>
  <Channels/>
</Container>
  ...
</DisplayProfile>
```

```
dp @ o=acme.com:
<DisplayProfile version="1.0" priority="10">
  ...
  <Container name="TemplateTableContainer" provider="TemplateTableContainerProvider"
merge="fuse">
  <Properties>
    ...
  </Properties>
  <Available>
    ...
  </Available>
  <Selected merge="replace" lock="false">
    <Reference value="Bookmark"/>
    <Reference value="Notes"/>
  </Selected>
  <Channels/>
</Container>
  ...
</DisplayProfile>
```

```
dp @ cn=hradmin,o=acme.com:
<DisplayProfile version="1.0" priority="5">
  ...
  <Container name="TemplateTableContainer" provider="TemplateTableContainerProvider"
merge="fuse">
  <Properties>
    ...
  </Properties>
  <Available>
  <Selected merge="fuse" lock="true">
    <Reference value="MailCheck"/>
    <Reference value="SampleSimpleWebService"/>
  </Selected>
  <Channels/>
</Container>
  ...
</DisplayProfile>
```

ディスプレイプロファイルドキュメントの優先順位設定の要約

ディスプレイプロファイルドキュメントの優先順位は、定義要因にマージ順序を考慮するか、ロック機能を考慮するかによって低くなったり高くなったりします。

ロックを考慮しない場合、低い数値のディスプレイプロファイルドキュメントは優先順位が低くなります。低い数値のディスプレイプロファイルドキュメントは最初にマージされるので、優先順位が低いドキュメントの値は優先順位が高いドキュメントの値で上書きされてしまいます。低い数値のドキュメントは優先順位が低くなります。

ただし、高い数値のドキュメントによる影響を受けないように、低い数値のディスプレイプロファイルドキュメントはオブジェクトをロックできます。この場合、低い数値のドキュメントは優先順位が高くなります。

ディスプレイプロファイルのマージ動作

ディスプレイプロファイルは、XML ドキュメント階層から構成されます。Portal Server はユーザー用のディスプレイプロファイルドキュメント、ユーザーが所属する各ロール、およびユーザーの組織やサブ組織を格納できます。実行時に、システムはこれらの複数のディスプレイプロファイルドキュメントをマージして、特定のポータルデスクトップをユーザーに配信します。このディスプレイプロファイルドキュメントのマージプロセスは、チャンネル、プロバイダ、プロパティ定義を変更する可能性があるため、最終的なディスプレイプロファイルに影響します。

ディスプレイプロファイルデータ形式には、これらのドキュメントの結合方法を定義する構文が収められています。この定義は、一般にマージ動作として知られています。

マージ動作は、異なる LDAP ノード (基本 DN、DN、ロール DN) のディスプレイプロファイルドキュメントをマージして単一の表現 (すなわちデスクトップ) を形成する場合に、属性の結合方法を制御します。マージ動作はディスプレイプロファイルドキュメントの順序付けを想定しています。Sun Java System Access Manager は、ロールの階層構造には対応していません。このため、ユーザーのロール構造はフラットです。すべてのロールは同位です。このため、Portal Server は Sun Java System Access Manager のロールに対して、階層構造をシミュレートするように追加の順序付けを行います。

ユーザーのディスプレイプロファイルドキュメントセットは、以下のドキュメントから構成されます。ユーザーの LDAP 組織およびサブ組織ノードに存在するドキュメント、ユーザーの各ロールノードに存在するドキュメント、ユーザーのエントリノードに存在するドキュメント。これらのノードのそれぞれにドキュメントを定義する必要はありませんが、ノードには少なくとも 1 つのドキュメントが定義されている必要があります。ドキュメントセットは、ディスプレイプロファイルドキュメントが定義する優先順位の値に従ってソートされます。詳細については、「[ディスプレイプロファイルドキュメントの優先順位](#)」を参照してください。

ディスプレイプロファイルドキュメントが順に積み重ねられるドキュメントのマージプロセスは、視覚化することができます。マージは名前の付いたチャンネル、プロバイダ、プロパティなどが重なり合う場所で起こります。マージはディスプレイプロファイルドキュメントで定義される XML 構造ではなく、ディスプレイプロファイルオブジェクトの名前に基づきます。名前の付いたチャンネルなどは、マージされるディスプレイプロファイルの格納階層内の異なるコンテナに存在できます。

マージプロセスの仕組み

ユーザーが Portal Server にログインし、認証が行われた後、システムは次の方法でユーザーのディスプレイプロファイルを判断します。

1. グローバルディスプレイプロファイル、およびユーザーが所属する組織、サブ組織、ロール、ユーザーの LDAP ノードを検索し、そのユーザーのすべてのディスプレイプロファイルドキュメントを探します。
2. 検索したディスプレイプロファイルドキュメントを一時領域に配置します。この領域をバッグ (入れ物) と想定します。
3. 優先順位に基づいてバッグ内のディスプレイプロファイルドキュメントを最下位から順にソートします (ドキュメントを検索するノードは優先順のソートに影響しない。また、ユーザーのディスプレイプロファイルドキュメントは常に最上位の順序になる)。
4. 最下位から順にバッグからドキュメントを取り出し、そのドキュメントの上に1つ上の順位のドキュメントを置き、マージとロック動作を適用します。
5. すべてのドキュメントがバッグから取り出されるまで手順 4 を繰り返し、ドキュメント内で見つかったすべてのオブジェクトのマージに等しい値が、システムからユーザーに返されるようにします。

ディスプレイプロファイルのマージタイプ

ディスプレイプロファイルは次の3種類のマージを使用して、ディスプレイプロファイルドキュメントの結合方式を決定します。

- **replace:** 優先順位の高いドキュメントで定義されたすべてのディスプレイプロファイルオブジェクトが低い順位のドキュメントで定義されたオブジェクトを完全に上書きします。優先順位の低いドキュメントにオブジェクトが存在しない場合、オブジェクトはマージ結果に追加されます (マージ結果の値は、オブジェクトによって置き換えられる)。
- **remove:** 名前が付けられたオブジェクトは、このポイントまでマージ結果から削除されます (マージ結果からオブジェクトが削除される)。これは、ディスプレイプロファイルに存在しなくなります (ただし、マージされる別のドキュメントによって持ち込まれる可能性はある)。これは、優先順位の高いドキュメントで再定義できます。
- **fuse:** 優先順位が低いドキュメントからのオブジェクトと、優先順位が高いドキュメントからのオブジェクトが組み合わせられます (オブジェクトは、マージ結果の値とマージされる)。

注 各マージタイプの正確な意味は、適用されるディスプレイプロファイルオブジェクトにより異なります。

チャンネルとプロバイダについては、fuse は特別な意味を持ちます。チャンネルそのものは、実際には融合されません。その代わりに、fuse はチャンネルまたはプロバイダのプロパティを結合する必要があることを指示します。replace 動作は、すべてのプロパティを含むチャンネルまたはプロバイダ全体を置換します。remove 動作は、そのポイントまでチャンネルまたはプロバイダ全体をマージから削除します。

ディスプレイプロファイル <DisplayProfile> ルートノードもマージ動作を持つことができます。replace 動作は、優先順位の高いドキュメントで定義されたすべての DP オブジェクトにより、優先順位の低いドキュメントで定義されたオブジェクトを完全に上書きすることを意味します。そのポイントまでのすべてのマージは無効になり、順位の高いドキュメントがマージの新しい基準として使用されます。remove 動作は、そのドキュメントのポイントまでのすべてのマージ結果が破棄されることを示します。マージはソートされたディスプレイプロファイルドキュメントセットの次のドキュメントから開始されます。チャンネルおよびプロバイダと同様に、fuse 動作はセット内のオブジェクト (チャンネルとプロバイダ) を結合する必要があることを意味します。

不可分のディスプレイプロファイルプロパティ (他のプロパティを格納できないプロパティ) は、fuse 動作を使用できません。これには文字列、整数、ブール型、参照の各プロパティが含まれます。

チャンネルのプロパティセットは、チャンネルのプロパティとチャンネルのプロバイダのプロパティ、およびチャンネルの親のプロパティなどから構成されます。プロパティ全体は、チャンネルの単一のドキュメントプロパティと考えることができます。ドキュメントのマージとは、あるドキュメントの全プロパティセットが、チャンネルの各ドキュメントのプロパティを結合したものから構成され、それらのプロパティはユーザーのマージセットのドキュメントすべてで使用されていることを意味します。

remove の例 : remove マージによるコンテナの選択チャンネルリストの修正

次の例は、すべてのユーザーのマージセットが、次のディスプレイプロファイルコード例を持つ組織レベルのドキュメントからどのように構成されるかを示します。

```

<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
  <Properties> ...</Properties>

  <Available> ...</Available>
  <Selected merge="fuse">
    <Reference value="UnixTipoftheDay"/>
  </Selected>
</Container>

```

「unix tip of the day」は、UNIX の使い方を説明しています。管理ロールに所属するユーザーは、このチャンネルがあまり有効ではないことに気が付いているようです。管理ロールを持つすべてのユーザーからこのチャンネルを削除するために、次のように管理ロールドキュメントで TemplateTableContainer チャンネルを定義します。

```

admin role
<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
  <Properties> ...</Properties>

  <Available> ...</Available>
  <Selected merge="fuse">
    <Reference value="Outages"/>
    <Reference value="SolarisAdmin"/>
    <Reference value="AdminTipoftheDay"/>
    <Reference value="UnixTipoftheDay" merge="remove"/>
  </Selected>
</Container>

```

上記のサンプルでは Reference value="UnixTipoftheDay" は、管理ロールディスプレイプロファイルから削除されます。

replace の例 : replace マージによるすべてのユーザーの表示からのチャンネルの削除

次の例は、特定のコンテナについて、ロール管理が組織レベルで定義されたすべてのチャンネルを無効にする方法を示しています。組織定義は次のようになります。

```
organization display profile
<Container name=...>
  ...
  ...
  <Selected>
    <Reference name="X"/>
    <Reference name="Y"/>
    <Reference name="Z"/>
  </Selected>
</Container>
```

ロール管理者はそのロールが割り当てられたどのユーザーにも X、Y、Z チャンネルを使用させたくないため、コンテナを次のように定義します。

```
admin role
<Container name=...>
  ...
  ...
  <Selected merge="replace">
    <Reference name="A"/>
    <Reference name="B"/>
    <Reference name="C"/>
  </Selected>
</Container>
```

ロールドキュメントのコンテナで選択されたリストは、組織のドキュメントのコンテナで選択されたリストと置換されます。

fuse の例 : fuse マージによるロールベースのチャンネルリストの作成

通常は fuse マージ動作を使用して、不可分でないディスプレイプロファイルオブジェクトを結合します。これらのオブジェクトには collection と使用可能なチャンネルリストまたは選択されたチャンネルリストが含まれます。ここでは、fuse は不可分でないプロパティに格納されたすべてのプロパティもマージする必要があることを示しています。このように fuse を使用することで、ユーザーに最終的に表示される不可分でないプロパティを、さまざまなドキュメントから構築することができます。

次の例のディスプレイプロファイルドキュメントは、管理、従業員、および movieFreak ロールに所属するユーザーに使用されます。最後に、ユーザーに選択されたチャンネルが表示されます。

```

admin role
<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
  <Properties> ...</Properties>

  <Available> ...</Available>
  <Selected merge="fuse">
    <Reference value="Outages"/>
    <Reference value="SolarisAdmin"/>
    <Reference value="AdminTipoftheDay"/>
  </Selected>
</Container>

```

```

employee role
<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
  <Properties> ...</Properties>

  <Available> ...</Available>
  <Selected merge="fuse">
    <Reference value="Benefits"/>
    <Reference value="EmployeeNews"/>
  </Selected>
</Container>

```

```

movieFreak role
<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
  <Properties> ...</Properties>

  <Available> ...</Available>
  <Selected merge="fuse">
    <Reference value="NewMoviesReleases"/>
    <Reference value="MovieShowTimes"/>
  </Selected>
</Container>

```

ユーザーに選択されたチャンネルの最終的なリストは次に示すとおりです。使用可能なチャンネルリストは、マージの適用順序と同様に、優先順位が低いものから高いものの順に並べられます。

```

<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
  <Properties> ...</Properties>

  <Available> ...</Available>
  <Selected merge="fuse">
    <Reference value="Outages"/>
    <Reference value="SolarisAdmin"/>
    <Reference value="AdminTipoftheDay"/>
    <Reference value="Benefits"/>
    <Reference value="EmployeeNews"/>
    <Reference value="NewMoviesReleases"/>
    <Reference value="MovieShowTimes"/>
  </Selected>
</Container>

```

マージロック

マージ可能なすべてのディスプレイプロファイルオブジェクトは、ロックすることもできます。オブジェクトをロックすると、そのオブジェクトは優先順位が高いドキュメントのマージ動作の影響を受けなくなります。これにより、優先順位の低いドキュメントでは、特定のディスプレイプロファイルが、優先順位の高いドキュメントのマージ動作によって変更されることを防ぐことができます。

マージロックの例 : lock マージによるすべてのユーザーに対するプロパティ値の強制

次の例は、特定の組織のすべてのユーザーに「employee news」チャンネルを表示する方法を示しています。ユーザーはこのチャンネルを表示から削除できません。組織レベルのドキュメントで、コンテナのチャンネルで選択されるリストは次のように定義されます。

```
<Selected merge="fuse">
  ...
  <Reference value="EmployeeNews" lock="true"/>
  ...
</Selected>
```

マージロックの例 : lock マージによるすべてのユーザー表示からのチャンネルの強制削除

次の例は、「online games」チャンネルを強制的に削除する方法を示しています。このシナリオでは、ユーザーが各ユーザードキュメントの選択したチャンネルリストにこのチャンネルを追加していた場合、組織レベルのドキュメントの選択したチャンネルリストからチャンネルを削除するだけでは機能しません。従業員と組織のリストを統合すると、「online games」チャンネルが表示されます。組織内のすべてのユーザーからチャンネルを強制的に削除するには、選択したチャンネルリストを次のように定義します。

```
<Selected merge="fuse">
  ...
  <Reference value="OnlineGames" merge="remove" lock="true"/>
  ...
</Selected>
```

remove 動作はマージされた結果からチャンネルを削除し、lock は優先順位の低いドキュメントの値がマージされるのを防ぎます。

ディスプレイプロファイルと Sun Java System Access Manager

ユーザーのディスプレイプロファイルドキュメントセットは、以下のドキュメントから構成されます。

- ユーザーの LDAP 組織 (またはサブ組織) ノードに存在するドキュメント
- 各ユーザーのロールノードに存在するドキュメント
- ユーザーのエントリノードに存在するドキュメント
- グローバルディスプレイプロファイルに存在するドキュメント

これらのノードのそれぞれにドキュメントを定義する必要はありませんが、ノードには少なくとも1つのドキュメントが定義されている必要があります。ドキュメントセットは、ディスプレイプロファイルドキュメントが定義する優先順位の値に従ってソートされます。詳細については、「[ディスプレイプロファイルドキュメントの優先順位](#)」を参照してください。マージ動作は、異なるノードのディスプレイプロファイルドキュメントをマージして単一の表示またはデスクトップを形成する場合に、属性を結合する方式を制御します。詳細については、「[ディスプレイプロファイルのマージ動作](#)」を参照してください。

管理者は Sun Java System Access Manager 管理コンソールを使用してディスプレイプロファイルを編集できます。Sun Java System Access Manager 管理コンソールでは、ディスプレイプロファイルが表示されないように、委任された管理者を設定することができます。この設定は、デスクトップサービステンプレートを作成する場合に行います。デスクトップサービスのテンプレートを作成するときに、「デスクトップサービスの属性を表示」ボックスにチェックマークを付けなければ、委任された管理者にディスプレイプロファイルテキストは表示されません。

ヒント 組織の管理者は、ポータルデスクトップサービスを通じて特定のロールと関連付けられたコンテナ (またはコンテナ階層) を定義できます。この場合、委任された管理者 (ロール管理者) は、「ポータルデスクトップ」属性ページの「チャンネルおよびコンテナ管理」リンクを通じて、このコンテナ内に必要なチャンネルとコンテナを定義できます。詳細については、「[「チャンネルおよびコンテナを管理」リンクを使用したチャンネルの管理](#)」を参照してください。

ディスプレイプロファイルの管理

ディスプレイプロファイルと、その他のポータルデスクトップサービスのデータは、Sun Java System Access Manager 管理コンソールと `dadmin` コマンドを利用して編集します。ディスプレイプロファイルを編集するときは、ディスプレイプロファイルのプロバイダ、コンテナ、およびチャネルを追加、修正、削除し、プロパティを編集します。

また、Sun Java System Access Manager 管理コンソールは、チャネルを追加し、既存のプロパティを編集するための「チャネルおよびコンテナを管理」リンクをポータルデスクトップ属性ページに表示します。このリンクを使用すると、チャネルを新規に作成したときに、プロパティを修正することもできます。

注 「チャネルおよびコンテナを管理」リンクは委任された管理者に合わせて調整され、管理者がコンテナおよびチャネルの属性をこのリンクで追加および修正できます。システム全体の管理者は、委任された管理者が使用できるコンテナおよびプロバイダを追加することに対して責任を持つ必要があります。

227 ページの表 10-2 は、ディスプレイプロファイルの各タイプと、Sun Java System Access Manager 管理コンソールを使用してそれを管理する方法を示しています。この 3 列の表では、最初の列はディスプレイプロファイルのタイプ、2 番目の列は Sun Java System Access Manager を使用してそのディスプレイプロファイルにアクセスする方法、3 番目の列は簡単な説明を示します。

表 10-2 ディスプレイプロファイルドキュメントのタイプ

ディスプレイプロファイルドキュメントのタイプ	Sun Java System Access Manager 管理コンソールを使用して表示する方法	説明
グローバルディスプレイプロファイルドキュメント	「サービス設定」を選択します。「ポータルデスクトップ」の隣にあるプロパティの矢印をクリックします。「グローバル」セクションで、「XML の直接編集」をクリックします。	属する組織またはロールに関係なく、システム上のすべてのユーザーに継承されるディスプレイプロファイル要素を定義します(現在は実施していません)。全員が使用する共通プロバイダを定義するために、ディスプレイプロファイルの XML ドキュメントを使用する必要があります(発生することもある)。

表 10-2 ディスプレイプロファイルドキュメントのタイプ (続き)

ディスプレイプロファイルドキュメントのタイプ	Sun Java System Access Manager 管理コンソールを使用して表示する方法	説明
ダイナミックディスプレイプロファイルドキュメント	「サービス設定」を選択します。「ポータルデスクトップ」の隣にあるプロパティの矢印をクリックします。「デスクトップダイナミック属性」セクションで、「XMLの直接編集」をクリックします。	コンテナ管理とチャネルのプロパティを説明します。このディスプレイプロファイルは、実行時にユーザーのデスクトップの生成には使用されませんが、新規に作成された各組織およびロールのデフォルトになります。デフォルトでは、ダイナミックディスプレイプロファイルドキュメントは空白です。ダイナミックディスプレイプロファイルを使用するには、まずそれを実装する必要があります。
組織、サブ組織、またはロールのディスプレイプロファイル	「アイデンティティ管理」を選択します。適切な組織、サブ組織を選択し、必要に応じて「表示」メニューから「ロール」を選択します。「表示」メニューから「サービス」を選択します。「ポータルデスクトップ」の隣にあるプロパティの矢印をクリックします。デスクトップページで、「XMLの直接編集」をクリックします。	<p>選択した組織、サブ組織、またはロールのディスプレイプロファイルを表示します。組織、サブ組織、ロールを新規作成した場合は、このエンティティのテンプレートを作成します。デスクトップサービスにテンプレートを作成する場合は、初期のディスプレイプロファイルは上記に述べたダイナミックディスプレイプロファイルドキュメントに設定されます。このため、ダイナミックディスプレイプロファイルが空白の場合は、何も指定されていません。</p> <p>ほとんどの場合、このディスプレイプロファイルドキュメントを使用して、コンテナ管理とチャネルのプロパティを異なる組織およびロールのニーズに合わせてカスタマイズします。</p>

Portal Server をインストールするときに、最初の組織を作成します。インストーラがディスプレイプロファイルグローバルレベルのドキュメントをインポートし、指定した内容に基づいてデフォルトのディスプレイプロファイルをインポートします。

その後は、組織、サブ組織、またはロールを新規に作成しても、ディスプレイプロファイルは毎回自動的にロードされません。新規に作成された組織、サブ組織、またはロールについてディスプレイプロファイルを手動でロードする必要があります。詳細については、「[ディスプレイプロファイルをロードする \(管理コンソール\)](#)」を参照してください。

ディスプレイプロファイルを管理するための手順の概要は、次のとおりです。

1. 新規に作成された組織、サブ組織、ロールのディスプレイプロファイルをロードします (インストールプロセスの間に作成される組織については、この手順を実行する必要はない)。

2. dpadmin コマンドや「XML の直接編集」リンクを使用するか、または「XML のダウンロード」と「XML のアップロード」リンクを使用して保存およびロードされたファイルとしてディスプレイプロファイルを修正します。
3. チャンネルとコンテナを追加して、「チャンネルおよびコンテナを管理」リンクを使用してそれらのプロパティを追加、削除、修正します。

デフォルトのディスプレイプロファイルドキュメント

表 10-3 は、サンプルポータルのインストール時に Portal Server デスクトップが /opt/SUNWps/samples/desktop ディレクトリに供給するディスプレイプロファイルドキュメントを示しています。次の表には 2 つの列があります。最初の列はディスプレイプロファイルドキュメント、2 番目の列はその説明を示します。

表 10-3 サンプルポータルで提供されるディスプレイプロファイルドキュメント

ディスプレイプロファイルドキュメント	説明
dp-anon.xml	認証なし匿名ユーザーによって使用されます。
dp-org.xml	デフォルトの組織レベルでロードされるサンプルディスプレイプロファイル。組織で使用される組織とチャンネルの定義のためのグローバルプロパティをすべて定義します。
dp-org-final.xml	dp-org.xml のコピーと、ブックマークとアプリケーションのチャンネルで定義される NetMail リンク。このディスプレイプロファイルドキュメントは、NetMail サービスが作成される時に使用されます。
dp-providers.xml	グローバルディスプレイプロファイルレベルでロードされるサンプルディスプレイプロファイル。このドキュメントは、すべてのプロバイダ定義を定義します。これらのプロバイダはすべての組織で使用されるため、システムは最上位でこのディスプレイプロファイルをロードし、すべての組織がそれらのプロバイダを使用できます。プロバイダ定義が 1 つの組織でのみ使用される場合、その組織レベルのディスプレイプロファイルでプロバイダを定義します。

これらのサンプルディスプレイプロファイルのカスタマイズについては、『Sun Java System Portal Server 6 2004Q2 Desktop Customization Guide』を参照してください。

ディスプレイプロファイルのロード

Portal Server を最初にインストールすると、インストーラが最初の組織を作成します。また、インストーラがディスプレイプロファイルグローバルレベルのドキュメントをインポートし、指定した内容に基づいてデフォルトのディスプレイプロファイルをインポートします。サンプルポータルをインストールしない場合、サンプルディスプレイプロファイルドキュメントはインストールされません。

その後、組織、サブ組織、またはロールを新規に作成したときに、ディスプレイプロファイルは自動的にロードされません。新規に作成された組織、サブ組織、またはロールについてディスプレイプロファイルを手動でロードする必要があります。

ディスプレイプロファイルのロードには、基本的に3つの方法があります。

- Sun Java System Access Manager 管理コンソールの「XML の直接編集」リンクを使用する。この方法では、「XML の直接編集」リンクと組織の既存のディスプレイプロファイルを使用します。このディスプレイプロファイルを、新規に作成した組織、サブ組織、またはロールの空のディスプレイプロファイルにコピーアンドペーストします。「[ディスプレイプロファイルをロードする \(管理コンソール\)](#)」を参照してください。
- コマンドラインを使用する。この方法では、dpadmin コマンドを使用してディスプレイプロファイルをロードします。「[ディスプレイプロファイルをロードする \(コマンド行\)](#)」を参照してください。dpadmin コマンドを使用する前に、「[dpadmin コマンドの使用に関するガイドライン](#)」を参照してください。
- Sun Java System Access Manager 管理コンソールの「XML をファイルにダウンロード」リンクと「XML をファイルからアップロード」リンクを使用する。この方法では、ディスプレイプロファイルをファイルにダウンロードし、ファイルからディスプレイプロファイルをアップロードします。「[ディスプレイプロファイルをダウンロードおよびアップロードする](#)」を参照してください。

注 Netscape 4.x ブラウザを使用している場合は、管理コンソールから直接ディスプレイプロファイルのXMLを編集できません。

ディスプレイプロファイルをロードする (管理コンソール)

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。

デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。

2. ディスプレイプロファイルをコピーする組織またはサブ組織を選択します。
3. 「表示」メニューから「サービス」を選びます。
4. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティの矢印を選択します。

データパネルにポータルデスクトップの属性が表示されます。

ヒント デスクトップサービスが表示されるまで、スクロールが必要なこともあります。

5. ディスプレイプロファイルをコピーします。
「XMLの直接編集」をクリックして、ディスプレイプロファイルのテキスト全体を選択してコピーします。
6. ディスプレイプロファイルをロードする組織、サブ組織、またはロールを選択します。
7. ナビゲーションパネルの「表示」メニューから「サービス」を選択します。
8. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティの矢印を選択します。

ディスプレイプロファイル XML を含む、ポータルデスクトップサービス属性の一覧がデータパネルに表示されます。

9. 「XMLの直接編集」をクリックします。
ディスプレイプロファイルの XML ドキュメントがテキストウィンドウに表示されます。
10. コピーしたディスプレイプロファイルをディスプレイプロファイルウィンドウにペーストします。
11. 完了したら、「保存」を選択します。

この変更は、特定の組織内のユーザーだけに反映されます。

ディスプレイプロファイルをロードする (コマンド行)

ディスプレイプロファイルをロードする場合は、dpadmin コマンドの modify サブコマンドを使用します。

たとえば、次のコマンドはディスプレイプロファイル (dp-org.xml) をロードします。

```
dpadmin add -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d  
"o=sesta.com,o=isp" dp-org.xml
```

注 -r または --dry-run オプションをファイル名の前のコマンドの最後に追加すると、実際に変更内容を LDAP に書き込む前に、コマンドが成功するかどうかを確認できます。

ディスプレイプロファイルをダウンロードおよびアップロードする

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ディスプレイプロファイルをコピーする組織またはサブ組織を選択します。
3. ナビゲーションパネルの「表示」メニューから「サービス」を選択します。
4. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティの矢印を選択します。
データパネルにポータルデスクトップの属性が表示されます。

ヒント デスクトップサービスが表示されるまで、スクロールが必要なこともあります。

5. 「グローバル属性」セクションの「XML のダウンロード」をクリックして、ファイルにディスプレイプロファイルを保存します。
6. ディスプレイプロファイルをアップロードする組織、サブ組織、またはロールをナビゲーションパネルで選択します。
7. ナビゲーションパネルの「表示」メニューから「サービス」を選択します。

8. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティの矢印を選択します。
9. 「XML のアップロード」をクリックして、ロードするファイルを指定します。
10. 「アップロード」をクリックします。
ディスプレイプロファイルのアップロードの完了を示すメッセージが表示され
ます。
11. 「閉じる」をクリックします。
この変更は、特定の組織内のユーザーだけに反映されます。

ディスプレイプロファイル全体を表示する

- `dpadmin list` コマンドと `list` サブコマンドを次のように実行して、ディスプレイプロファイル全体を表示します。

```
dpadmin list -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d  
"o=sesta.com,o=isp"
```

ディスプレイプロファイルを削除する

破損した場合など、何らかの理由でディスプレイプロファイルを削除する必要がある場合は、`dpadmin` コマンドと `remove` サブコマンドを使用します。

たとえば、ルートからディスプレイプロファイル (`dp-org.xml`) 全体を削除する場合、次のように指定します。

```
dpadmin remove -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d  
"o=sesta.com,o=isp" -t root
```

ルートから、またはディスプレイプロファイルが必要なノードからディスプレイプロファイルを削除する場合、新しいディスプレイプロファイルをロードする必要があります。たとえば、上記のように `dp-org.xml` ディスプレイプロファイルを削除した場合、`dp-org-final.xml` ディスプレイプロファイルのように、類似した別のディスプレイプロファイルをロードする必要があります。ディスプレイプロファイルのロードについては、「[ディスプレイプロファイルをロードする \(コマンド行\)](#)」を参照してください。

「チャンネルおよびコンテナを管理」リンクを使用したチャンネルの管理

管理は「チャンネルおよびコンテナを管理」リンクを使用して行います。

- プロパティ: グローバルディスプレイプロファイルのプロパティを定義および追加できます。
- コンテナ: コンテナのコンテンツコンテナを追加または削除できます。また、コンテンツコンテナのプロパティを修正することもできます。
- チャンネル: コンテナのチャンネルを追加または削除できます。
- チャンネルのプロパティを修正することもできます。

注 現在、「チャンネルおよびコンテナを管理」リンクを使用して、チャンネルとコンテナ、およびそれらのプロパティを操作することができます。このリンクでプロバイダの操作は実行できません。

デスクトップの属性ページを使用する場合、委任された管理者には「チャンネルおよびコンテナを管理」リンクのみが表示されます。他のすべてのディスプレイプロファイルの属性は非表示になっているので安全です。

チャンネルとコンテナ管理のデフォルトプロバイダ

「デスクトップ」の「チャンネルおよびコンテナを管理」リンクには、コンテナチャンネルまたはコンテンツチャンネルを追加または削除ができる管理画面が表示されます。

チャンネルの追加

「チャンネル」リストの「追加」リンクを使用して、定義されているコンテンツプロバイダのリストからコンテンツプロバイダを選択し、それを追加することができます。[234 ページの表 10-4](#) は、新規チャンネルの作成時に基本として使用できるプロバイダチャンネルを示しています。この表では、最初の列はプロバイダ名、2 番目の列はその簡単な説明を示します。定義されているコンテンツプロバイダについては、『Sun Java System Portal Server 6 2005Q1 Desktop Customization Guide』を参照してください。

表 10-4 定義されているプロバイダチャンネル

プロバイダ	説明
AppProvider	Web アプリケーションにリンクのリストを表示します (ユーザーはこのリストをカスタマイズできる)。

表 10-4 定義されているプロバイダチャンネル (続き)

プロバイダ	説明
BookmarkProvider	ポータルページに表示されるブックマークのリストをユーザーが管理できます。
JSPProvider	1 つまたは複数の JSP™ ファイルからコンテンツを取得します。
LoginProvider	ユーザーは匿名ポータルページから Sun Java System Access Manager への認証を行えます。
MailCheckProvider	ユーザーのメールステータスに関する情報を表示します。
NotesProvider	システム全体のメッセージをリスト表示し、このようなメッセージをユーザーが送信できるようにします。
SearchProvider	Sun Java System Portal Server の検索エンジンを使用した検索機能を提供します。
URLScrapperProvider	指定した URL からコンテンツを取得し、Sun Java System Portal Server を使用してコンテンツをフォーマットします。
UserInfoProvider	ディスプレイプロファイルと Access Manager から情報を収集します。あいさつ、ユーザー名、タイムゾーン、ロケールが表示され、ユーザーの IMAP データと SMTP データにアクセスできます。
XMLProvider	指定した URL から XML コンテンツを取得し、XSLT を使用してコンテンツをマークアップ言語に変換します。

シンプル Web サービスプロバイダ

シンプル Web サービス (SWS) プロバイダは、データ指向の Web サービスにアクセスする機能を提供します。このプロバイダに基づいて、サンプルチャンネルは通貨変換サービスにアクセスする Web サービスの実装を示します。

シンプル Web サービスチャンネルには次の 2 種類があります。

- [事前に設定されている Web サービスチャンネル](#)
- [新規コンテナチャンネル](#)

デフォルトでは、事前に設定されている Web サービスチャンネルのサンプルがサンプルポータルデスクトップに用意されています。管理者は、設定可能な Web サービスチャンネルのサンプルを Access Manager 管理コンソールから追加できます。

どちらの Web サービスチャンネルも、複雑な入力パラメータやユーザーインターフェース表示要件を持たない Web サービスのように、比較的シンプルな Web サービスでの使用に最適です。特定の Web サービスを処理できないことをシンプル Web サービスプロバイダが検出すると、それに応じてメッセージがユーザーに表示されます。

このプロバイダに基づくチャンネルが一度にバインドできるのは、1つの Web サービスと関連メソッドだけです。シンプル Web サービスプロバイダは、整数、文字列、ダブルなど、シンプルなデータタイプをサポートします。このリリースのシンプル Web サービスプロバイダは、次の機能を備えています。

- 入力および出力パラメータ内のシンプルおよび複雑なタイプの配列をサポートする。
- NestedComplex Type、1 次元で同種の SimpleType、および ComplexType の配列をサポートする。
- Java では配列、構造体 (Java Beans)、SimpleType (整数、文字、文字列) の 1 次元配列、および構造体 / オブジェクト (Complex Type) に相当する、複雑な xml メッセージを含む Web サービスをサポートする。
- Web サービスにおける一方向操作をサポートする。Web サービスにおいて、通知形式の一方向操作のみをサポートする。送信請求 / 応答形式のサービスはサポートしません (現在の jax-rpc 標準の制限)。
- WSDL 定義内のバインド処理において、障害データの使用をサポートしない。

シンプル Web サービスプロバイダは、次の WSDL 設定プロパティタイプをサポートします。

- SOAP バインドスタイル: RPC、ドキュメント
- SOAP エンコードタイプ: 符号化済み、リテラル

注 RPC とリテラルの組み合わせはサポートされません。 .Net ベースの Web サービスのサポートは制限されます。

事前に設定されている Web サービスチャンネル

事前に設定されている Web サービスチャンネルのサンプルを使用すると、通貨変換サービスに接続できます。

事前に設定されている Web サービスチャンネルを設定するには、管理コンソールから WSDL URL とメソッド名を指定する必要があります。

設定可能な Web サービスチャンネル

設定可能な Web サービスチャンネルを使用することで、ユーザーは指定の Web サービスをポイントするようにチャンネルを変更できます。これは、WSDL URL の値と、Web サービスに関連付けられているメソッド名をユーザーが変更できるようにすることで可能になりました。ただし、事前に設定されているチャンネルタイプとは異なり、設定可能な Web サービスチャンネルには、Web サービス入力パラメータのデフォルト値を格納する機能がありません。

新規コンテナチャンネル

「コンテナチャンネル」リストの「新規」リンクを使用して、定義されているコンテナプロバイダのリストからコンテナプロバイダを選択し、それを作成することができます。表 10-5 は、新規チャンネルの作成時に基本として使用できる、定義済みのプロバイダチャンネルを示しています。この表では、最初の列はプロバイダ名、2 番目の列はその簡単な説明を示します。定義されているコンテンツプロバイダについては、『Sun Java System Portal Server 6 2004Q2 Desktop Customization Guide』を参照してください。

表 10-5 定義されているプロバイダコンテナチャンネル

プロバイダ	説明
JSPFrameCustomTableContainerProvider	ユーザーの JSP フレームセットベースのポータルデスクトップにフレームを新規作成します。
JSPSingleContainerProvider	単一のチャンネルを表示します。
JSPTabContainerProvider	それぞれがタイトルを持つ複数のタブから構成されるチャンネルを表示します。
JSPTabCustomTableContainerProvider	ユーザーの JSP タブベースのデスクトップにタブを新規作成します。
JSPTableContainerProvider	テーブルにコンテンツチャンネルを表示します。
TemplateEditContainerProvider	「編集」ページのフレームを描画します。
TemplateTabContainerProvider	複数タブをサポートします。
TemplateTabCustomTableContainerProvider	タブを新規作成します。
TemplateTableContainerProvider	テーブルにコンテンツチャンネルを表示します。

チャネルまたはコンテナチャネルを作成する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. チャネルを追加する組織、サブ組織、またはロールを選択します。
委任された管理者としてログインした場合、管理者としてアクセスできる組織、サブ組織、またはロールに自動的に移動します。
3. ナビゲーションパネルの「表示」メニューから「サービス」を選択します。
4. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティの矢印を選択します。
デスクトップ属性ページがデータパネルに表示されます。
5. デスクトップページで「チャネルおよびコンテナを管理」リンクを選択します。
「チャネル」ページが表示され、コンテナパスが **root** で設定されています。
6. チャネルまたはコンテナを追加するコンテナをクリックします。
ページの上部に、チャネルが追加されるコンテナパスが表示されます。チャネルとコンテナが定義されている場合は、リストに表示されます。
7. 「新規」をクリックして、コンテナチャネルまたはチャネルを追加します。
コンテナチャネルを追加するには、「コンテナチャネル」の下の「新規」をクリックします。チャネルを追加するには、「チャネル」の下の「新規」をクリックします。
「新規チャネル」ページが表示されます。
8. チャネル名を入力し、メニューからプロバイダのタイプを選択します。
使用可能なプロバイダについては、[234 ページの表 10-4](#) を参照してください。
9. 「作成」をクリックします。

チャンネルまたはコンテナチャンネルのプロパティを変更する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. チャンネルを修正する組織、サブ組織、またはロールを選択します。
委任された管理者としてログインした場合、管理者としてアクセスできる組織、サブ組織、またはロールに自動的に移動します。
3. 「表示」メニューから「サービス」を選びます。
4. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティの矢印を選択します。
デスクトップ属性ページがデータパネルに表示されます。
5. デスクトップページで「チャンネルおよびコンテナを管理」リンクを選択します。
「チャンネル」ページが表示されます。ページの上部にコンテナのパスが表示されます。定義済みのチャンネルがリストに表示されます。
6. 変更するチャンネルまたはコンテナチャンネルの横の「プロパティの編集」リンクを選択します。
「プロパティ」ページが表示されます。
7. 必要に応じてプロパティを修正します。
チャンネルのプロパティについては、『Sun Java System Portal Server 6 2004Q2 Desktop Customization Guide』を参照してください。
8. 完了したら、「保存」を選択します。

チャンネルまたはコンテナチャンネルを削除する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. チャンネルを修正する組織、サブ組織、またはロールを選択します。
委任された管理者としてログインした場合、管理者としてアクセスできる組織、サブ組織、またはロールに自動的に移動します。
3. ナビゲーションパネルの「表示」メニューから「サービス」を選択します。
4. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティの矢印を選択します。
デスクトップ属性ページがデータパネルに表示されます。
5. デスクトップページで「チャンネルおよびコンテナを管理」リンクを選択します。
「チャンネル」ページが表示されます。ページの上部にコンテナのパスが表示されます。定義済みのチャンネルがリストに表示されます。
6. 削除するチャンネルまたはコンテナチャンネルの隣にあるチェックボックスをクリックします。「削除」をクリックします。
7. チャンネルが削除され、「チャンネル」リストが更新され、削除が確認できます。

コンテナの管理

コンテナを管理する場合、Sun Java System Access Manager 管理コンソールを使用してディスプレイプロファイルの XML を直接編集することができます。また dpadmin コマンドを使用することもできます。このコマンドはこの節のほとんどの箇所、さまざまな例を使用して説明しています。

これには、次の例が含まれます。

- [ディスプレイプロファイルオブジェクトを表示する](#)
- [コンテナのチャンネルを置換する](#)
- [チャンネルのプロパティを置換する](#)

- コンテナにチャンネルを追加する
- コレクションにプロパティを追加する
- コレクションプロパティを追加する
- チャンネルまたはコンテナからプロパティを削除する
- プロバイダを削除する
- コンテナからチャンネルを削除する
- ディスプレイプロファイルドキュメントの優先順位を変更する
- チャンネルをコンテナで使用できるようにする
- チャンネルをコンテナで使用不可にする
- コンテナの使用可能なチャンネルリストからチャンネルを選択する
- コンテナの使用可能なチャンネルリストからチャンネルを選択解除する

Sun Java System Access Manager 管理コンソールによるディスプレイプロファイルの編集については、「ディスプレイプロファイルテキストウィンドウの使用」を参照してください。

dpadmin コマンドの使用

dpadmin コマンドの構文は次のとおりです。

```
$ dpadmin list|merge|modify|add|remove [command-specific options] -u uid -w password
{-g|-d dn} [-l locale] [-r] [-b] [-h] {-v|--version} [file]
```

dpadmin コマンドの構文に関する詳細については、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。dpadmin コマンドを実行する場合は、次の点に注意してください。

- file 引数: 指定するときは、file 引数はコマンドラインの最後の引数として指定する必要があります。この引数は、ディスプレイプロファイル DTD に適合する XML コードを含む XML ファイルへのパスを指定します。XML の入力が必要なサブコマンドには、modify および add が含まれます。

ディスプレイプロファイル全体を追加または修正する場合、常に適切な XML ヘッダーを追加します。

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
```

- **list** サブコマンド: ディスプレイプロファイルノードオブジェクトを取得し、表示します。オブジェクトはネイティブ XML 形式で表示されます。表示されるオブジェクトは、標準出力に送信されます。-n または -name オプションを使用しない場合、ディスプレイプロファイルドキュメント全体が表示されます。-n または -name オプションが DP ノードオブジェクトを指定しない場合、DP ドキュメント全体が表示されます。
- **merge** サブコマンド: 指定した DP ノードオブジェクトのマージ結果を取得し、表示します。オブジェクトはネイティブ XML 形式で表示されます。表示されるオブジェクトは、標準出力に送信されます。-n または -name オプションを使用しない場合、エラーが出力されます。
- **modify** サブコマンド: 既存のディスプレイプロファイルオブジェクトの値を変更します。このコマンドは、オブジェクトがすでにディスプレイプロファイルに存在することを想定しています。modify サブコマンドは、標準入力または引数として指定されたファイルのいずれかから、新規オブジェクトのデータを読み出します。新しいオブジェクト用のデータは、XML であり、ディスプレイプロファイル DTD に適合する必要があります。特に、オブジェクトデータは正しくフォーマットされた XML コードにする必要があります。
- **add** サブコマンド: ディスプレイプロファイルに新しいオブジェクトを追加します。このサブコマンドは、追加されるオブジェクトがディスプレイプロファイルに存在しないことを前提としています。add サブコマンドは、標準入力または引数として指定されたファイルのいずれかから新規オブジェクトのデータを読み出します。新しいオブジェクト用のデータは、XML であり、ディスプレイプロファイル DTD に適合する必要があります。特に、オブジェクトデータは正しくフォーマットされた XML コードにする必要があります。
- **remove** サブコマンド: ディスプレイプロファイルから既存のオブジェクトを削除します。
- **-g** オプション: グローバルレベルのディスプレイプロファイルドキュメントを指定します。
- **-d dn** オプション: dpadmin を実行する DN を指定します。-d オプションと -g オプションは相互に排他的です。
- **-r** または **--dry-run** オプション: 現在のコマンドの成功または失敗を示し、変更を LDAP に書き込みません。これは、特定の dpadmin コマンドのフォーマットが正しいことを確認する場合に便利です。

- `-n` または `--name` オプション: ディスプレイプロファイルコンテナ、チャンネルまたはプロバイダオブジェクト、またはディスプレイプロファイルオブジェクトの親の、完全修飾名を指定します。name 引数が DP ノードオブジェクトを指定しない場合、エラーが出力されます。
- `-p` または `--parent` オプション: 親ディスプレイプロファイルコンテナ、チャンネルまたはプロバイダオブジェクト、またはディスプレイプロファイルオブジェクトの親の、完全修飾名を指定します。
- `-v` または `--version` オプション: `dpadmin` コマンドのバージョン番号を、標準出力に出力します。

dpadmin コマンドの使用に関するガイドライン

`dpadmin` コマンドを使用してディスプレイプロファイルを更新する場合は、次のガイドラインに従います。

- 他の管理者が同時に、Sun Java System Access Manager 管理コンソールまたは `dpadmin` コマンドを使用して、ディスプレイプロファイルの修正を行っていないことを確認します。このような状況では、`dpadmin` コマンドと管理コンソールが同時にディスプレイプロファイルにアクセスするのを防ぐロックの仕組みが機能しないため、変更内容が損なわれてしまいます。
- `dpadmin` を使用する場合、XML「コード」としてファイルに修正内容を保存し、`dpadmin` コマンドと `add` サブコマンドを実行する方法を推奨します。たとえば次のように指定します。

```
/opt/SUNWps/bin/dpadmin add -u
"uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d
"uid=anonymous,ou=people,o=sesta.com,o=isp" newtheme.xml
```

この例で、`newtheme.xml` は、ディスプレイプロファイルに追加される XML「コード」を含むファイルです。

- ディスプレイプロファイルドキュメントを直接編集する場合、まず `dpadmin` コマンドと `list` サブコマンドを使用してディスプレイプロファイルの最新の内容を取得します。編集したら、`dpadmin` コマンドと `modify` サブコマンドを実行します。たとえば次のように指定します。

```
/opt/SUNWps/bin/dpadmin list
-u"uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d
"o=sesta.com,o=isp" > dp-org.xml
```

(`dp-org.xml` ファイルを編集)

```
/opt/SUNWps/bin/dpadmin modify -u
"uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d
"o=sesta.com,o=isp" dp-org.xml
```

警告 dpadmin list と dpadmin modify コマンドを実行する間に、(管理コンソール、dpadmin、または ldapmodify コマンドを使用する) どのような方法でも LDAP サーバーでディスプレイプロファイルドキュメントを変更しないでください。変更した場合、その変更内容は最新の dpadmin modify によって上書きされます。

ディスプレイプロファイルの修正

ディスプレイプロファイルオブジェクトは、次のいずれかを実行して修正できます。

- 既存のディスプレイプロファイルドキュメントを手動で編集し、dpadmin modify コマンドを使用して適切な LDAP ノードまたはグローバルレベルでロードします。
- 標準入力で、XML テキストの変更内容を指定して、dpadmin コマンドを実行します。新規オブジェクトを追加する場合は、add サブコマンドを使用します。既存のオブジェクトを修正する場合は、modify サブコマンドを使用します。
- 新規のディスプレイプロファイルドキュメントを最初から作成し、dpadmin modify コマンドを使用して、適切な LDAP ノードまたはグローバルレベルでロードします。

ディスプレイプロファイルのエラーメッセージの概要

無効な XML を含むディスプレイプロファイルドキュメントを作成し、保存しようとする、システムがエラーを出力します。このエラーメッセージは、タイトル、メッセージ、およびサブメッセージとして表示されます。メッセージボックスのタイトルは、「無効な XML ドキュメント」です。メッセージは、次のいずれかの形式で表示されます。

- XML のパースに失敗しました。
- XML の doctype がありません。
- DP の保存に失敗しました。
- 無効な XML 入力です。

「無効な XML ドキュメント」エラーが表示された場合、エラーを修正して、作業中の XML ドキュメントを保存する必要があります。

ディスプレイプロファイルオブジェクトを表示する

- `list` サブコマンドを使用してディスプレイプロファイルオブジェクトを表示します。

たとえば、次のコマンドは `TemplateTableContainer` というチャンネル、コンテナ、またはプロバイダを取得し、標準出力に出力します。

```
dpadmin list -n "TemplateTableContainer" -u
"uid=amAdmin,ou=people,o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp"
```

注 `-n` オプションを省略すると、ディスプレイプロファイルドキュメント全体を表示できます。

コンテナのチャンネルを置換する

1. `modify` サブコマンドを使用して、コンテナ内のチャンネルを標準入力で指定した値に置換します。

たとえば、このコマンドはコンテナ `TemplateTableContainer` のチャンネル `Test` を標準入力で指定された値に置換します。

```
dpadmin modify -p TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" <<EOF
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Channel name="Test" provider="testprovider">
  <Properties>
    <String name="title" value="Test Channel"/>
    <String name="description" value="This channel is a test."/>
  </Properties>
</Channel>
EOF
```

2. `list` サブコマンドを使用して、チャンネルが置換されたことを確認します。

詳細については、「[ディスプレイプロファイルオブジェクトを表示する](#)」を参照してください。

チャンネルのプロパティを置換する

1. `modify` サブコマンドを使用して、チャンネル内のプロパティを標準入力で指定した値に置換します。

たとえば、次のコマンドはチャンネル `NewChannel` で動作し、プロパティ `new.xml` を前述のファイルの新規オブジェクトと置換します。この場合 `new.xml` は次のようになります。

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<String name="welcome" value="Hi, welcome to your desktop!"/>
```

```
dpadmin modify -p TemplateTableContainer/NewChannel -u
"uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp"
new.xml
```

2. `list` サブコマンドを使用して、プロパティが置換されたことを確認します。

詳細については、「[ディスプレイプロファイルオブジェクトを表示する](#)」を参照してください。

コンテナにチャネルを追加する

1. ディスプレイプロファイル入力 XML ファイルを修正して、新規の <Channel> 定義のみを追加し、次のようにファイル testadd.xml を作成します。

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Channel name="TestChannel" provider="testprovider">
  <Properties>
    <String name="teststring" value="sfds"/>
  </Properties>
</Channel>
```

2. add サブコマンドを使用して、コンテナにチャネルを追加します。

たとえば、次のコマンドは testadd.xml で定義された新規チャネルをディスプレイプロファイルに追加します。次の例では、新規チャネルを TemplateTableContainer レベルに追加する必要があります。親オブジェクトを -p オプションで指定しない場合、チャネルはルートレベルで追加されます。

```
dpadmin add -p "TemplateTableContainer" -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" testadd.xml
```

注 新規チャネルを JSPTabContainer に追加する場合、実際には新規タブを追加しています。JSPTabContainer では、使用可能で選択されたすべてのタブに TabProperties が定義されている必要があります。このため、JSTabContainer に追加された新規コンテナまたはチャネルについて、新規チャネルまたはコンテナが追加された JSPTabContainer の TabProperties Collection 内に次の XML コードを追加します。

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Collection name="<New Channel Name>">
  <String name="title" value="<New Channel Title>"/>
  <String name="desc" value="<New Channel Description>"/>
  <Boolean name="removable" value="false"/>
  <Boolean name="renamable" value="true"/>
  <Boolean name="predefined" value="true"/>
</Collection>
```

3. `list` サブコマンドを使用して、チャンネルが追加されたことを確認します。
詳細については、「[ディスプレイプロファイルオブジェクトを表示する](#)」を参照してください。

コレクションにプロパティを追加する

1. `combine (-m)` オプションを使用して、コレクションに新規プロパティを追加します。
たとえば、次のコマンドは新規プロパティ `msg2` をコレクション `bar` に追加します。既存のプロパティ `msg` は、まだ残っています。`list` サブコマンドは、プロパティ値を表示する前後に使用されます。

```
dpadmin list -n TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp"
...
<Collection name="foo">
  <Collection name="bar">
    <String name="msg" value="hi"/>
  </Collection>
</Collection>
...
```

```
dpadmin modify -p TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" -m <<EOF

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Collection name="foo">
  <Collection name="bar">
    <String name="msg2" value="woo hoo"/>
  </Collection>
</Collection>
EOF
```

```

dpsadmin list -n TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp"
...
<Collection name="foo">
  <Collection name="bar">
    <String name="msg" value="hi"/>
    <String name="msg2" value="woo hoo"/>
  </Collection>
</Collection>
...

```

コレクションプロパティを追加する

1. add サブコマンドを使用して、標準入力で指定した値を持つコレクションを追加します。

たとえば、次のコマンドは標準入力で指定されたコレクションプロパティ zipCodes を Postal というチャンネル、コンテナ、またはプロバイダに追加します。

```

dpsadmin add -p SamplesTabPanelContainer/Postal -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" <<EOF
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Collection name="zipCodes">
  <Integer value="98012"/>
  <Integer value="98036"/>
  <Integer value="94025"/>
  <Integer value="95112"/>
</Collection>
EOF

```

2. list サブコマンドを使用して、コレクションプロパティが追加されたことを確認します。

詳細については、「[ディスプレイプロファイルオブジェクトを表示する](#)」を参照してください。

チャンネルまたはコンテナからプロパティを削除する

1. `remove` サブコマンドを使用して、チャンネルまたはコンテナからプロパティを削除します。

たとえば、次のコマンドはプロパティ `locations` をグローバルレベルで `Bookmarks` チャンネル (またはコンテナ) から削除します。

```
dpadmin remove -t property -p Bookmarks -n locations -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -g
```

2. `list` サブコマンドを使用して、プロパティが削除されたことを確認します。

詳細については、「[ディスプレイプロファイルオブジェクトを表示する](#)」を参照してください。

プロバイダを削除する

1. `remove` サブコマンドを使用してプロバイダを削除します。

たとえば、次のコマンドはプロバイダ `NotesProvider` を削除します。

```
dpadmin remove -t provider -n "NotesProvider" -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp"
```

2. `list` サブコマンドを使用して、プロバイダが削除されたことを確認します。

詳細については、「[ディスプレイプロファイルオブジェクトを表示する](#)」を参照してください。

コンテナからチャネルを削除する

1. `remove` サブコマンドを使用して、コンテナからチャネルを削除します。

たとえば、次のコマンドは、`TemplateTableContainer` という親コンテナ内に存在する `Test` というチャネルを削除します。

```
dpadmin remove --type channel --parent TemplateTableContainer --name "Test"
--runasdn "uid=amAdmin,ou=People,o=sesta.com,o=isp" --password password --dn
"o=sesta.com,o=isp"
```

2. `list` サブコマンドを使用して、チャネルが削除されたことを確認します。

詳細については、[第 10 章「ディスプレイプロファイルの管理」](#) を参照してください。

ディスプレイプロファイルドキュメントの優先順位を変更する

1. `modify` サブコマンドを使用して、ディスプレイプロファイルドキュメントの優先順位を変更します。

たとえば、次のコマンドはドキュメントの優先順位を元の順序から組織の 10 に変更します。

```
dpadmin modify -m -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d
"o=sesta.com,o=isp" <<EOF
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<DisplayProfile priority="30" version="1.0"
<Properties/>
<Channels/>
<Providers/>
EOF
```

2. `list` サブコマンドを使用して、優先順位が変更されたことを確認します。

詳細については、「[ディスプレイプロファイルオブジェクトを表示する](#)」を参照してください。

チャンネルをコンテナで使用できるようにする

1. `modify` サブコマンドと `combine (-m)` オプションを使用して、標準入力指定されたチャンネルをコンテナの既存の使用可能リストに追加します。

たとえば、次のコマンドは BookMark チャンネルを `TemplateTableContainer` の Available リストに追加します。

```
dpadmin modify -p TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" -m <<EOF

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Available>
  <Reference value="BookMark">
</Available>
EOF
```

2. `list` サブコマンドを使用して、優先順位が変更されたことを確認します。

詳細については、「[ディスプレイプロファイルオブジェクトを表示する](#)」を参照してください。

チャンネルをコンテナで使用不可にする

1. `remove` サブコマンドを使用して、コンテナの Available リストからチャンネルを削除します。

たとえば、次のコマンドは、`TemplateTableContainer` という親コンテナの Available リストから `Test` というチャンネルを削除します。

```
dpadmin remove --type available --parent TemplateTableContainer --name "Test"
--runasdn "uid=amAdmin,ou=People,o=sesta.com,o=isp" --password password --dn
"o=sesta.com,o=isp"
```

2. `list` サブコマンドを使用して、チャンネルが削除されたことを確認します。

詳細については、「[ディスプレイプロファイルオブジェクトを表示する](#)」を参照してください。

コンテナの使用可能なチャンネルリストからチャンネルを選択する

1. `modify` サブコマンドと `combine (-m)` オプションを使用して、標準入力指定されたチャンネルをコンテナの既存の `Selected` リストに追加します。

たとえば、次のコマンドは `BookMark` チャンネルを `TemplateTableContainer` の `Selected` リストに追加します。

```
dpadmin modify -p TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" -m <<EOF

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Selected>
  <Reference value="BookMark">
</Selected>
EOF
```

2. `list` サブコマンドを使用して、優先順位が変更されたことを確認します。

詳細については、「[ディスプレイプロファイルオブジェクトを表示する](#)」を参照してください。

コンテナの使用可能なチャンネルリストからチャンネルを選択解除する

1. `remove` サブコマンドを使用して、コンテナの `Selected` リストからチャンネルを削除します。

たとえば、次のコマンドは、`TemplateTableContainer` という親コンテナの `Selected` リストから `Test` というチャンネルを削除します。

```
dpadmin remove --type selected --parent TemplateTableContainer --name "Test"
--runasdn "uid=amAdmin,ou=People,o=sesta.com,o=isp" --password password --dn
"o=sesta.com,o=isp"
```

2. `list` サブコマンドを使用して、チャンネルが削除されたことを確認します。

詳細については、「[ディスプレイプロファイルオブジェクトを表示する](#)」を参照してください。

ディスプレイプロファイルテキストウィンドウの使用

Sun Java System Access Manager はディスプレイプロファイルテキストを表示し、直接編集するためのテキストウィンドウを提供します。組織、サブ組織、またはロールに管理者としてアクセスできる間は、Sun Java System Access Manager 管理コンソールを使用して、このテキストウィンドウに移動し、ディスプレイプロファイルを表示または編集できます。

注 Netscape 4.x ブラウザを使用している場合は、管理コンソールから直接ディスプレイプロファイルの XML を編集できません。

ディスプレイプロファイルテキストウィンドウにアクセスする

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ディスプレイプロファイルドキュメントを修正する組織、サブ組織、またはロールを選択します。
委任された管理者としてログインした場合、管理者としてアクセスできる組織、サブ組織、またはロールに自動的に移動します。
3. ナビゲーションパネルの「表示」メニューから「サービス」を選択します。
4. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティの矢印を選択します。
デスクトップ属性ページがデータパネルに表示されます。

5. デスクトップページで、「XMLの直接編集」リンクをクリックします。
ディスプレイプロファイルがテキストウィンドウに表示されます。

注 デフォルトでは、ディスプレイプロファイルの優先順位レベルはキーワード `user` に設定され、現在のディスプレイプロファイルがユーザーレベルのディスプレイプロファイルであることを示します。その他の許容される値は、低い優先順位を表す小さい整数です。たとえば、1は2よりも優先順位が低くなります。

6. 変更したら、「保存」を選択します。

注 グローバル、組織、サブ組織、またはロールレベルのドキュメントへの変更は、即時有効になります。ユーザーレベルのドキュメントへの変更は、ユーザーがログアウトおよびログインした後で有効になります。

NetMail サービスの管理

NetMail サービスを管理する方法について説明します。この章で説明する内容は次のとおりです。

- [NetMail サービスの概要](#)
- [NetMail サービスの管理](#)

NetMail サービスの概要

NetMail サービスは、電子メールクライアントの NetMail (Java™) および NetMail Lite を実装しています。これらのクライアントは、標準の IMAP および SMTP サーバー環境で動作します。NetMail によって、ユーザーは 1 つまたは複数のメールサーバーにアクセスし、電子メールの読み取り、作成、削除、およびフォルダの作成、アクセス、削除を行うことができます。

Sun Java™ System Portal Server 6 では、Sun Java™ System Access Manager の管理コンソールを利用して、NetMail サービスの定義および管理を行います。NetMail サービスは、NetMail クライアントが電子メールメッセージとその設定を管理するために、サービス属性とデフォルト値を定義します。管理者は、組織および組織のユーザーに対してサービス属性の値を定義およびカスタマイズし、NetMail クライアントの動作方式を制御します。

NetMail サービスの管理

Sun Java System Access Manager のポリシーサービスにより、ルールを定義したり、リソースにアクセスしたりすることができます。ポリシーは、ロールまたは組織に基づき、アクセス権を与えたり、制約を定義したりすることができます。

デフォルトでは、ポリシー設定サービスは自動的に最上位の組織に追加されます。サブ組織は、親組織とは別にポリシーサービスを追加する必要があります。作成したポリシーサービスは、すべての組織に追加する必要があります。ポリシーを使用するための手順の概要は、次のとおりです。

1. 組織のポリシーサービスを追加します。
2. サブ組織の参照ポリシーを作成します。組織のポリシーに関する定義と決定を別の組織に委託することができます。または、リソースのポリシー定義を別のポリシー製品に委託することもできます。この、ポリシーの作成と評価に関するポリシー委任は、参照ポリシーによって制御されます。これは、ルールと参照自体から構成されます。リソースを必要としないアクションがポリシーサービスに含まれる場合、サブ組織の参照ポリシーを作成することはできません。詳細については、[260 ページの「サブ組織の参照ポリシーを作成する」](#)を参照してください。
3. ピアまたはサブ組織の標準のポリシーを作成します。アクセス権の定義には標準のポリシーを使用します。標準のポリシーは、複数のルール、サブジェクト、条件から構成されます。詳細については、[261 ページの「サブ組織に標準のポリシーを作成する」](#)を参照してください。

ピアまたはサブ組織のポリシーサービスを追加する

サブ組織は、親サービスを継承しないため、サブ組織のポリシーサービスを追加する必要があります。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 参照ポリシーを作成する組織またはサブ組織に移動します。

作成したすべての組織がナビゲーションパネルに表示されます。

3. ナビゲーションパネルの「表示」メニューから「組織」を選択し、「名前」メニューから適切な組織を選択します。

4. 「表示」メニューから「サービス」を選択します。

5. 「追加」をクリックします。

「サービスを追加」ページがデータパネルに表示されます。NetMail サービスのチェックボックスにチェックマークを付けて、「了解」をクリックします。

新規に追加されたサービスがナビゲーションパネルに表示されます。

6. プロパティの矢印をクリックして NetMail サービスを設定します。

7. データパネルに、次のメッセージが表示されます。

「現在このサービスにはテンプレートが存在しません。新規に作成しますか？」

8. メッセージボックス内の「作成」をクリックし、テンプレートを作成します。

NetMail の属性が表示されます。

9. NetMail の属性を変更します。

NetMail 属性についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

10. 「保存」をクリックして、変更後の値をサービステンプレートに保存します。

注 新しい組織を作成する場合は、その組織用の NetMail ポリシーを作成して割り当てる必要があります。サンプルポータルの場合、すでにデフォルトで NetMail が有効になっているため、この作業は必要ありません。

サブ組織の参照ポリシーを作成する

組織のポリシーに関する定義と決定を別の組織に委託することができます。この、ポリシーの作成と評価に関するポリシー委任は、参照ポリシーによって制御されます。これは、ルールと参照自体から構成されます。参照は、ルール内のリソースとして親組織を定義する必要があり、参照内の値として `SubOrgReferral` に組織名が含まれている必要があります。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 参照ポリシーを作成するサブ組織を含む組織に移動します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「表示」メニューから「ポリシー」を選択します。
4. 「新規」をクリックし、新しいポリシーを作成します。
ポリシーの作成ページがデータパネルに表示されます。
5. 「名前」には、`SubOrgReferral_NetMail` を入力します。「ポリシータイプ」で「参照」が選択されていることを確認します。「作成」をクリックします。
6. データパネルの「表示」メニューから「ルール」を選択し、「追加」をクリックします。「NetMail」が選択されていることを確認し、「次へ」をクリックします。
「ルールを追加」テンプレートがデータパネルに表示されます。
7. 「サービス」から `NetMail` を選択し、「次へ」をクリックします。
8. 「ルール名」に `NetMailRule` と入力し、「作成」をクリックします。
9. データパネルの「表示」メニューから「参照」を選択し、「追加」をクリックします。
「参照を追加」テンプレートがデータパネルに表示されます。
10. 「名前」に `SubOrgReferral_suborg_name` と入力します。
データパネルで「値」にサブ組織の名前が選択されていることを確認し、「作成」をクリックしてポリシーの設定を完了します。
11. データパネルで「保存」をクリックします。

データが保存されると、「ポリシープロパティが保存されました。」というメッセージが表示されます。

サブ組織に標準のポリシーを作成する

アクセス権の定義には標準のポリシーを使用します。標準のポリシーは、複数のルール、サブジェクト、条件から構成されます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. ポリシーを作成する組織またはサブ組織に移動します。
作成したすべての組織がナビゲーションパネルに表示されます。
3. 「表示」メニューから「ポリシー」を選択します。
その組織のポリシーが表示されます。
4. ナビゲーションパネルで「新規」を選択します。「新規ポリシー」ページがデータパネルに表示されます。
5. 「名前」に SubOrgNormal_NetMail と入力します。「ポリシータイプ」で「標準」が選択されていることを確認します。「作成」をクリックします。
6. データパネルの「表示」メニューから「ルール」を選択し、「追加」をクリックします。「ルールを追加」テンプレートがデータパネルに表示されます。
7. 「サービス」メニューから「NetMail」を選択し、「次へ」をクリックします。「ルール名」に NetMailRule と入力します。「NetMail 実行権限を所有」にチェックマークが付けられていることを確認します。
8. 「サービス」メニューから「NetMail」を選択し、「次へ」をクリックします。「NetMail 実行権限を所有」にチェックマークが付けられていることを確認します。
9. 「タイプ」メニューからサブジェクトのタイプを選択し、「次へ」をクリックしてサブジェクトの設定を完了します。
10. データパネルの「表示」メニューから「サブジェクト」を選択し、「追加」をクリックします。「サブジェクトを追加」ページがデータパネルに表示されます。

11. 「作成」をクリックして、ポリシーの設定を完了します。

データが保存されると、「ポリシープロパティが保存されました」というメッセージが表示されます。

NetMail のサービス属性の変更 (特定の組織)

NetMail のサービスをカスタマイズするには、サービスの属性を変更します。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 組織を選択します。
3. 「表示」メニューから「サービス」を選びます。
4. ナビゲーションパネルの「NetMail」の隣にあるプロパティの矢印をクリックします。
Netmail のサービス属性のリストが、データパネルに表示されます。
5. サービス属性の値を変更し、「保存」をクリックして変更を保存します。
この変更は、選択された組織内のユーザーだけに反映されます。
NetMail 属性についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

NetMail のサービス属性の変更 (すべての組織)

時として、グローバル NetMail サービスの属性値、すなわち、後で NetMail サービスに追加する組織すべてに反映させる属性値を修正する必要があります。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. ロケーションパネルで「サービス設定」をクリックします。
3. ナビゲーションパネルの「NetMail」の隣にあるプロパティの矢印を選択します。
NetMail のサービス属性のリストが、データパネルに表示されます。
4. サービス属性の値を変更し、「保存」を選択して変更を保存します。
この変更は、今後、NetMail サービスに追加するすべての組織に影響します。

新しいウィンドウを開くように NetMail Lite を設定する

デフォルト設定では、NetMail Lite が実行されている状態で、ユーザーがデスクトップにある NetMail Lite のリンクを選択してメッセージを作成している場合、NetMail Lite の現在のウィンドウは NetMail Lite の新しいインスタンスに置き換えられるため、メッセージのテキストは消滅します。この問題を避けるためには、NetMail Lite の設定を変更し、デスクトップにある NetMail Lite のリンクをユーザーがクリックするたびに、新しいウィンドウが開くようにします。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 組織を選択します。
3. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティ矢印を選択します。
デスクトップのサービス属性のリストが、データパネルに表示されます。
4. データパネルで、「チャンネルおよびコンテナを管理」のリンクをクリックします。
5. 「チャンネル」の下にある、App チャンネルの「編集」リンクをクリックします。

6. 組織を選択し、「表示」メニューから「サービス」を選択します。
7. targets プロパティの、「編集」リンクを選択します。
8. NetMail Lite のプロパティを、次のように書き換えます。

```
NetMail Lite | ^javascript:var nmServerURL = document.URL; nmDestURL
=nmServerURL.split('dt')[0];nmAdjustedURL = nmDestURL
+'NetMailServlet?nsid=newHTMLSession';
openAppURL(nmAdjustedURL, '_blank');return false;
```

9. 「保存」をクリックします。
10. 変更を確認します。

組織内のテストユーザーとしてログインします。NetMail Lite にアクセスし、メッセージの作成を開始します。NetMail Lite のリンクをクリックします。NetMail Lite を表示したウィンドウが新しく開かれます。

リモートアドレス帳の使用 (LDAP)

NetMail のリモートアドレス帳の機能を有効にするには、NetMail サービスの LDAP サーバーリストの属性を変更します。

注	検索エンジンが「含む文字列」、「一致」、「始めの文字列」、「終わりの文字列」、および「類似」の検索条件をサポートしている場合、アドレス帳の検索機能は、これらの検索条件で、ユーザーが指定したテキストを基に比較して名前を検索することができます。 個人のアドレス帳は、「含む文字列」の検索だけをサポートしています。LDAP アドレス帳を追加した場合は、そのほかのオプションが有効になります。
---	---

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。

デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルでは「組織」が選択されています。
2. 組織を選択します。
3. 「表示」メニューから「サービス」を選びます。
4. ナビゲーションパネルの「NetMail」の隣にあるプロパティの矢印をクリックします。

NetMail のサービス属性のリストが、データパネルに表示されます。

5. 「アドレス帳検索で使用する LDAP サーバーの詳細」の値を変更します。各エントリは、カンマで区切られた `name="value"` のペアのリストです。
- `name`: NetMail の「アドレス」ページに表示される名前 (デフォルトは、なし)
 - `server`: LDAP サーバーの完全修飾ドメイン名 (デフォルトは、なし)
 - `base`: 検索を開始するために使用する識別名 (DN) (デフォルトは、`""`)
 - `searchin`: カンマで区切られた検索する属性のリスト (デフォルトは、`"cn,gn,sn"`)
 - `result`: 電子メールアドレスを含む属性 (デフォルトは、`"mail"`)
 - `filter`: 検索に適用する追加の LDAP フィルタ (デフォルトは、`""`)。フィルタの構文には、LDAP フィルタ構文が使用されます。
 - `referral`: LDAP 参照を実行するかどうかを定義する値。デフォルトは `"follow"`。参照を実行しないように定義するには `""` を指定します。

たとえば、Sesta LDAP のディレクトリを検索するには、次のエントリを使用します。

```
name="Sesta  
LDAP",server="ldap-server.sesta.com",base="dc=sesta,dc=com"
```

6. 「保存」をクリックします。

リライターサービスの管理

この章では、Sun Java System Portal Server のリライターサービスの管理方法について説明します。

この章で説明する内容は次のとおりです。

- [リライターサービスの概要](#)
- [サポートされる URL](#)
- [リライタルールとルールセットの定義](#)
- [リライターサービスの管理](#)

リライターサービスの概要

Sun Java System Portal Server のリライタは、マークアップ言語と JavaScript™ コードで URL の変換を行うためのエンジンを提供します。デスクトップと Sun Java™ System Portal Server には URLScrapperProvider と XMLProvider があります。Secure Remote Access ゲートウェイサービスは、すべてリライターサービスを使用します。

リライタは Web ページの内容をスキャンし、それらの Web ページ上で検出した URL を特定します。リライタはルールセットで定義されるルールの集合を使用して、Web ページでリライトする要素を判断します。リライタが URL を特定すると、次の方法で URL をリライトします。

- [相対 URL の絶対 URL への拡張](#)
- [既存 URL へのゲートウェイ URL の追加](#)

相対 URL の絶対 URL への拡張

URLScrapperProvider はコア Portal Server 製品の一部を構成します。ゲートウェイを使用しないシナリオでは、URLScrapperProvider を使用して相対 URL を絶対 URL に拡張することができます。たとえば、ユーザーが次のサイトにアクセスするとします。

```
<a href=" ../mypage.html" >
```

リライターはこれを次のように解釈します。

```
<a href=" http://www.yahoo.com/mail/mypage.html" >
```

この `http://yahoo.com/mail/` は収集されたページのベース URL です。

URLScrapperProvider の制限

URLScrapperProvider は指定された URL のチャンネルの表示のみを試みます。ドキュメント URL (ドキュメント) の表示箇所を指定する方法はありません。

URLScrapperProvider は指定された URL の内容を要求するという点で、HTTP クライアントと同様に機能します。ブラウザの場合と同じく、収集のターゲットとなる URL はネットワーク上でアクセス可能か、プロキシを設定している必要があります。

最終的な URL スクレイパーチャンネルは、ミニブラウザでもフレームでもありません。したがって、コンテンツにリンクがある場合、チャンネルだけではなくページ全体に影響します。URL スクレイパーチャンネルの内部を参照することはできません。チャンネル内でリンクを選択すると、ブラウザはリンクを変換し、現在表示されているページ (ポータルサーバーのデスクトップ) をリンク場所のコンテンツに置き換えます。

収集されたチャンネルの外観は、オリジナルのコンテンツの作成元によって制御されます。URLScrapperProvider は全くコンテンツを修正せず、URL を通じて入手できる内容だけを表示します。チャンネルは本来 HTML テーブル内のセルであるため、テーブルセルに表示できる HTML コンテンツを表示するだけです。すなわち、URLScrapperProvider を使用してフレームセットを収集することはできません。これは <FRAMESET> タグを <BODY> タグ内で表示できないためです。URLScrapperProvider はまた、<HEAD> タグ内で JavaScript コードを実行しません。このため、次の収集シナリオは URLScrapperProvider については適切ではありません。

- ユーザーがチャンネルをカスタマイズできるように、ある種の編集機能が要求される場合。
- データが HTML 以外、Web 以外のサーバーのソース、すなわちデータベースかメールサーバーから導出されている場合。
- データをチャンネルに合った何らかの方法で再フォーマットする必要がある場合。
- URLScrapperProvider がそれぞれのデスクトップ表示およびユーザーに対して要求を行い、検索する際に、より効率的なソリューションが要求される場合。

元サーバーから cookie が送信される場合、その cookie は Web コンテンツが再収集されるたびに元サーバーに転送されます。このため、ポータルデスクトップが更新または再読み込みされたときに、初回の Web コンテンツの収集時に送信された cookie を元サーバーが取得することになります。しかし、ユーザーが URL スクレイパーチャネル内のリンクをクリックした場合は、これらの cookie は戻されません。

既存 URL へのゲートウェイ URL の追加

Sun Java System Portal Server Secure Remote Access などのゲートウェイを実装している場合、ゲートウェイはクライアントのプロキシとして動作し、イントラネットサイトにアクセスして、応答をクライアントに返します。リライタはダウンロードされたページで URL を変換し、既存の URL の前にゲートウェイ URL をプレフィックスとして指定して URL が元のサイトではなくゲートウェイを示すようにします。

たとえば、ユーザーが次の URL を使用して mymachine で HTML ページにアクセスしようとする場合、次のように指定します。

```
<a href=" http://mymachine.intranet.com/mypage.html">
```

リライタは、次のようにプレフィックスとしてゲートウェイを参照する URL を指定します。

```
<a href="https://gateway.company.com/http://mymachine.intranet.com/mypage.html">
```

ユーザーがこのアンカーに関連するリンクをクリックすると、ブラウザはゲートウェイに照会します。ゲートウェイは、mymachine.intranet.com から mypage.html のコンテンツをフェッチします。

リライタを使用して既存の URL のプレフィックスにゲートウェイ URL を指定する詳細については、『Sun Java System Portal Server Secure Remote Access 6 2005Q1 管理ガイド』を参照してください。

サポートされる URL

リライタは RFC-1738 で指定されるすべての標準 URL のリライトをサポートします。これらの URL はプロトコルが HTTP か HTTPS のいずれの場合も、プロトコルの大文字小文字に関係なくサポートされます。たとえば、hTtP、HTtp、および httP はすべて有効です。一部のサンプルの標準 URL を次に示します。

```
http://www.my.sesta.com
http://www.example.org:8000/imaginary/test
http://www.example.edu/org/admin/people#andy
http://info.example.org/AboutUs/Index/Phonebook?dobbins
http://www.example.org/RDB/EMP?*%20where%20name%%3Ddobbins
http://info.example.org/AboutUs/Phonebook
http://user:password@example.com
```

リライタは標準以外でも基本的な URL の一部については、リライトをサポートします。標準以外の URL を標準形式に変換するための情報は、URL が表示されるページの基本 URL から取り出され、プロトコル、ホスト名、パスを含めることができます。円記号 (¥) は、絶対 URL ではなく相対 URL に含まれる場合のみサポートされます。たとえば、http://sesta.com¥index.html はリライトされますが、http:¥¥sesta.com はリライトされません。

また、http://sesta.com のようにプロトコルまたはスキーマの後に単一のスラッシュ (/) が付いた URL はリライトされません。

リライタルールとルールセットの定義

リライタは Web ページに表示されるさまざまな要素の URL 部分を修正します。リライタには一連のデフォルトルールが付属し、これに基づいて Web ページのリライトする要素を決定します。さまざまなカテゴリとサブカテゴリのルールの集合は、.dtd ファイルに保存され、ルールセットと呼ばれます。リライタルールセットは XML で定義されます。

DTD は /opt/SUNWps/web-src/WEB-INF/lib/rewriter.jar (resources/RuleSet.dtd) に置かれています。ルールセットは URL を特定するために使用されます。デフォルトでは、"/"、"./"、"http"、および "https" などの文字で始まる Web コンテンツの文字列はすべて URL と見なされ、リライタの候補になります。

実装でリライタを設定する場合、ルールセットを作成し、管理コンソールで「Portal Server 設定」の「リライタ」セクションにルールを定義します。ルールセットの作成と修正については、「[リライターサービスの管理](#)」を参照してください。Web ページのコンテンツの種類に基づいて、複数のルールを定義します。たとえば、HTML コンテンツをリライトするのに必要なルールは、JavaScript コンテンツをリライトするためのルールと異なります。リライタのルールは、次の広範囲なカテゴリに分類されます。

- [HTML コンテンツのルール](#)
- [JavaScript コンテンツのルール](#)
- [XML コンテンツのルール](#)

注 Wireless Markup Language (WML) は HTML に類似し、HTML ルールは WML コンテンツに適用されます。

 CSS コンテンツにはルールが不要です。

ルールセットは XML ドキュメントであり、ドキュメント内の XML は正しく作成する必要があります。ルールセットでルールを定義する場合、次のガイドラインを守ってください。

- すべてのルールは `<ruleset>` `</ruleset>` タグで囲む必要があります。
- ルールセットの `<HTML>` `</HTML>` セクションに、HTML コンテンツをリライトするためのすべてのルールを指定します。
- ルールセットの `<JSRules>` `</JSRules>` セクションに、JavaScript コンテンツをリライトするためのすべてのルールを指定します。
- ルールセットの `<XML>` `</XML>` セクションに、XML コンテンツをリライトするためのすべてのルールを指定します。

HTML コンテンツのルール

Web ページの HTML コンテンツは、さらに属性、JavaScript トークン、フォーム、アプレットに分類できます。これに従って、HTML コンテンツのルールは次のように分類されます。

- [HTML コンテンツの属性ルール](#)
- [HTML コンテンツの JavaScript トークンルール](#)
- [HTML コンテンツのフォームルール](#)
- [HTML コンテンツのアプレットルール](#)

HTML コンテンツの属性ルール

属性ルールは、HTML ページでリライトする基本的な属性タグを特定します。リライトは既存の URL を拡張またはプレフィックスをつけて、定義済みのタグをさまざまな形に修正します。デフォルトのルールセットは、次の属性タグをリライトします。

- action
- background
- codebase
- code
- href
- src
- value
- imagePath
- lowsrc
- archive

属性ルールの構文は次のとおりです。

```
<Attribute name="name" [tag="tag" valuePatterns="patterns"]
```

ここで *name* は属性を指定します。*tag* は属性が所属するタグ (すべてのタグに一致させるには * を設定) を指定します。*patterns* は属性と一致する可能性のあるパターンを指定します。*tag* と *valuePatterns* パラメータはオプションです。

HTML コンテンツの JavaScript トークンルール

Web ページの JavaScript タグに純粋な JavaScript コードを記述することができます。あるいは JavaScript のトークンまたは関数を指定することができます。たとえば、Web ページに別の URL にジャンプする `onClick()` 関数を指定することができます。ページが正しく機能するために、`onClick()` 関数の値を変換し、リライトする必要があります。大半の場合、デフォルトのルールセットで指定されたルールで、JavaScript トークンの URL を十分にリライトできます。デフォルトのルールセットは、次の JavaScript トークンをリライトします。

- `onAbort`
- `onBlur`
- `onChange`
- `onClick`
- `onDblClick`
- `onError`
- `onFocus`
- `onKeyDown`
- `onKeyPress`
- `onKeyUp`
- `onLoad`
- `onMouseDown`
- `onMouseMove`
- `onMouseOut`
- `onMouseOver`
- `onMouseUp`
- `onReset`
- `onSelect`
- `onSubmit`
- `onUnload`

JavaScript トークンルールの構文は次のとおりです。

```
<JSToken>javascript_function_name</JSToken>
```

ここで `javascript_function_name` は `onLoad` や `onClick` などの関数名です。

HTML コンテンツのフォームルール

ユーザーはフォームを格納した HTML ページを参照できます。input などのフォーム要素は、URL を値として見なします。デフォルトのルールセットは、フォーム要素をリライトしません。フォームルールの構文は次のとおりです。

```
<Form source="/source.html" name="form1" field="field1">
[valuePatterns="pattern"] />
```

ここで `/source.html` はフォームを含む HTML ページの URL です。`form1` はフォームの名前、`field1` はリライトするフォームのフィールド、`pattern` はフィールドのリライトする部分を示します。指定されたパターンに従うすべてのコンテンツはリライトされます。

`valuePatterns` パラメータはオプションです。

HTML コンテンツのアプレットルール

単一の Web ページに複数のアプレットが含まれていたり、また各アプレットに多くのパラメータが指定されていたりすることがあります。アプレットの URL のリライタルールには、次に説明するようにパターンマッチング情報を含める必要があります。

- ソース、`filename.htm` など
- コード、`classname.class` など
- パラメータ名、`servername` など
- パラメータ値、`some_url` など

リライタは、ルールに指定された値をアプレットの内容と照合し、必要に応じて URL を修正します。この置換は、ユーザーが特定の Web ページを参照している場合ではなく、サーバーで行われます。値の一部にワイルドカード (*) も使用できます。たとえば、パラメータ名が * であれば、リライタはアプレットでパラメータ名を比較しません。

デフォルトのルールセットは、アプレットパラメータをリライトしません。

アプレットルールの構文は次のとおりです。

```
<Applet source="sourcehtml.jsp" code="class" param="parameter_name"
[valuePatterns="pattern"]
```

ここで `/sourcehtml.jsp` はアプレットを含む URL です。`class` はアプレットクラスの名前、`parameter_name` は値をリライトする必要があるパラメータ、`pattern` はフィールドのリライトする箇所を示します。指定されたパターンに従うすべてのコンテンツはリライトされます。`valuePatterns` パラメータはオプションです。

JavaScript コンテンツのルール

JavaScript コードのさまざまな部分に URL を指定することができます。リライタは JavaScript コードを直接パースできませんが、URL 部分を判断します。JavaScript プロセッサで URL の解釈を助けるために、一部の特別なルールを記述する必要があります。

URL を含む JavaScript 要素は次のように分類されます。

- [JavaScript 変数](#)
- [JavaScript 関数パラメータ](#)

JavaScript 変数

JavaScript 変数はさらに 5 つのカテゴリに分類されます。

- [JavaScript URL 変数](#)
- [JavaScript EXPRESSION 変数](#)
- [JavaScript DHTML 変数](#)
- [JavaScript DJS \(ダイナミック JavaScript\) 変数](#)
- [JavaScript システム変数](#)

JavaScript URL 変数

URL 変数の右側には URL 文字列が表示されます。デフォルトのルールセットは、次の JavaScript URL 変数をリライトします。

- `imgsrc`
- `location.href`
- `_fr.location`
- `mf.location`
- `parent.location`
- `self.location`

JavaScript コンテンツルールの URL 変数の構文は次のとおりです。

```
<Variable type="URL">variable_name</Variable>
```

ここで `variable_name` はリライトされる変数の名前です。

JavaScript EXPRESSION 変数

EXPRESSION 変数の右側には式が表示されます。この式の結果が URL です。リライタはこのような式を評価できないため、式を HTML ページに変換するための JavaScript 関数を付加します。この関数はパラメータに式をとり、クライアントブラウザで式を評価します。

デフォルトのルールセットは、JavaScript EXPRESSION 変数 `location` をリライトします。

JavaScript コンテンツルールの EXPRESSION 変数の構文は次のとおりです。

```
<Variable type="EXPRESSION">variable_exp</Variable>
```

ここで `variable_exp` は式変数です。

JavaScript DHTML 変数

DHTML は、HTML コンテンツを保持する JavaScript 変数です。デフォルトのルールセットは、次の JavaScript DHTML 変数をリライトします。

- `document.write`
- `document.writeln`

JavaScript コンテンツルールの DHTML 変数の構文は次のとおりです。

```
<Variable type="DHTML">variable</Variable>
```

ここで `variable` は DHTML 変数です。

JavaScript DJS (ダイナミック JavaScript) 変数

DJS (ダイナミック JavaScript) 変数は、JavaScript コンテンツを保持する JavaScript 変数です。

JavaScript コンテンツルールの DJS 変数の構文は次のとおりです。

```
<Variable type="DJS">variable</Variable>
```

ここで `variable` は DJS 変数です。

変数内の JavaScript コードは、変換するルールがもう一つ必要です。

JavaScript システム変数

システム変数はユーザーによって宣言されない変数ですが、JavaScript 標準の一部として使用できます。

デフォルトルールセットは `window.location.pathname` JavaScript システム変数をリライトします。

JavaScript コンテンツのシステム変数の構文は次のとおりです。

```
<Variable type="SYSTEM">variable</Variable>
```

ここで *variable* はシステム変数です。

JavaScript 関数パラメータ

関数パラメータは 4 つのカテゴリに分類されます。

- [JavaScript URL パラメータ](#)
- [JavaScript EXPRESSION パラメータ](#)
- [JavaScript DHTML パラメータ](#)
- [JavaScript DJS パラメータ](#)

JavaScript URL パラメータ

URL パラメータは、URL を直接含む文字列パラメータです。

デフォルトのルールセットは、次の JavaScript URL パラメータをリライトします。

- `openURL`
- `openAppURL`
- `openNewWindow`
- `parent.openNewWindow`
- `window.open`

URL パラメータの構文は次のとおりです。

```
<Function type = "URL" name = "function" [paramPatterns="y,y,"] />
```

ここで *function* は評価される関数の名前、*y* はリライトの必要があるパラメータの位置を示します。パラメータの位置は、カンマで区切られます。たとえば、構文の行で最初と 2 番目のパラメータはリライトする必要がありますが、3 番目のパラメータはリライトできません。

JavaScript EXPRESSION パラメータ

EXPRESSION パラメータは、評価後に URL になる関数内の変数です。EXPRESSION パラメータの構文は次のとおりです。

```
<Function type = "EXPRESSION" name = "function" [paramPatterns="y,y,"] />
```

ここで *function* は評価される関数の名前、*y* はリライトの必要があるパラメータの位置を示します。パラメータの位置は、カンマで区切られます。たとえば、構文の行で最初と 2 番目のパラメータはリライトする必要がありますが、3 番目のパラメータはリライトできません。

JavaScript DHTML パラメータ

DHTML パラメータは、HTML ページを動的に生成するネイティブ JavaScript メソッドです。たとえば、`document.write()` メソッドはこのカテゴリに分類されます。

デフォルトのルールセットは、次の JavaScript DHTML パラメータをリライトします。

- `document.write`
- `document.writeln`

DHTML パラメータの構文は次のとおりです。

```
<Function type = "DHTML" name = "function" [paramPatterns="y,y,"] />
```

ここで *function* は評価される関数の名前、*y* はリライトの必要があるパラメータの位置を示します。パラメータの位置は、カンマで区切られます。たとえば、構文の最初と 2 番目のパラメータはリライトする必要がありますが、3 番目のパラメータはリライトされません。

JavaScript DJS パラメータ

HTML の Cascading Style Sheets (CSS) などの動的 JavaScript (DJS) パラメータも変換されます。この変換に定義されるルールはありません。これは URL が CSS の `url()` 関数にのみ表示されるためです。DJS パラメータの構文は次のとおりです。

```
<Function type = "DJS" name = "function" [paramPatterns="y,y,"] />
```

ここで *function* は評価される関数の名前、*y* はリライトの必要があるパラメータの位置を示します。パラメータの位置は、カンマで区切られます。たとえば、構文の最初と 2 番目のパラメータはリライトする必要がありますが、3 番目のパラメータはリライトされません。

XML コンテンツのルール

Web ページには XML コンテンツが含まれ、XML には URL が含まれます。リライタは XML コンテンツの URL をリライトできます。

URL を含む XML コンテンツは、次のように分類されます。

- [XML のタグテキスト](#)
- [XML の属性](#)

XML のタグテキスト

リライタはタグ名に基づいて XML コンテンツを変換します。

デフォルトのルールセットは、次の XML のタグをリライトします。

- baseroot
- img

タグテキストの構文は次のとおりです。

```
<TagText tag ="attribute" attributePatterns="name=src"/>
```

ここで *attribute* はタグの名前、*src* は属性の名前です。

XML の属性

XML の属性のルールは、HTML の属性のルールに類似しています。詳細については、[272 ページの「HTML コンテンツの属性ルール」](#)を参照してください。リライタは属性とタグ名に基づいて属性の値を変換します。

デフォルトのルールセットは、次の XML の属性をリライトします。

- xmlns
- href

HTML の属性の構文は次のとおりです。

```
<Attributes>  
  <Attribute name="attribute" [valuePatterns="name=src"/>  
</Attributes>
```

ここで *attribute* はタグの名前、*src* は属性の名前です。

リライターサービスの管理

Portal Server 6 では、リライターサービスは Sun Java System Access Manager 属性を使用して、リライタルールセットの持続的ストレージを提供しています。リライタルールセットは、Web ページ内のコンテンツをリライターでどのようにリライトするかを定義します。Sun Java System Access Manager 管理コンソールを使用して、複数のリライタルールセットを Sun Java System Access Manager サービス属性値として定義および保存できます。

コマンドラインを使用してリライターを管理することもできます。radmin コマンドについての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

Sun Java System Access Manager 管理コンソールは、リライタルールセットの概念を持たないため、Portal Server はカスタマイズされたサービス管理プラグインモジュールを使用して、リライタルールセットを管理します。すべてのリライタルールセットは、Sun Java System Access Manager の組織に対してグローバルです。特定の組織レベルでのルールセットの作成はできません。

注 URLScrapperProvider は、HTML テーブルセル内で有効なコンテンツのみを収集できます。収集する HTML マークアップに、<body>、<base>、特定の JavaScript プロシージャなどのような、テーブルセル内で表示できないマークアップが含まれていると、「デスクトップ」ページは正常に表示されない場合があります。収集するコンテンツを定義する場合、コンテンツが有効な HTML であることを確認するようにしてください。詳細は、「[URLScrapperProvider の制限](#)」を参照してください。

リライター URLScrapperProvider を SSL 用に設定する

リライターの URLScrapperProvider を使用して、SSL ページを収集し、安全なセッションでアクセスできるように URL をリライトできます。

1. Portal Server をインストールしたサーバーに対して、信頼のできるデータベースを Web サーバー管理コンソールで次のように初期化します。
 - a. ブラウザから、次の URL を入力して Web サーバー管理ページにアクセスします。
http://servername:8088
 - b. 管理者としてログインし、「セキュリティ」タブをクリックします。
 - c. データベースのパスワードを 2 度入力し、「了解」を選択します。
2. 次の手順でパスワードファイルを作成します。

- a. ディレクトリを `/AccessManager-base/SUNWam/config` に変更します。
 - b. 非表示のテキストファイル `.wtpass` を作成します。
 - c. 信頼できるデータベースを初期化したときに指定したパスワードを入力します。
3. URLScraperProvider を使用してアクセスした Web サーバーに使用される証明書に、ルート CA がインストールされていない場合、
`/AccessManager-base/SUNWam/lib/AMConfig-instance_nickname.properties` ファイルに次の行を追加します。

```
com.sun.am.jssproxy.trustAllServerCerts=true
```

このオプションは、証明書が信用できることを JSS に示します。
 4. Portal Server を再起動します。

デフォルトテンプレートからルールセットを新規作成する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「リライター」の隣にあるプロパティの矢印をクリックします。
現在定義されている属性のリストが、データパネルに表示されます。
4. 「新規」をクリックします。
これによって、修正が可能なルールセットのテンプレートが表示されます。
5. `<RuleSet id="ruleset_template">` 行を編集し、`default_ruleset` を新しいルールセットの名前で置き換えます。
6. ルールセットテンプレート内のルールを追加または修正し、必要に応じて URL をリライトします。
7. 「作成」をクリックし新しいルールセットを作成します。

成功した場合、最初のページと現在定義されているすべてのルールセットのリストが表示されます。これには作成したルールセットが含まれています。

既存のルールセットを編集する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「リライト」の隣にあるプロパティの矢印をクリックします。
現在定義されている属性のリストが、データパネルに表示されます。
4. 編集するルールセットの「編集」リンクをクリックします。
編集するルールセットの XML が表示されます。
5. ルールセットテンプレート内のルールを追加または修正し、必要に応じて URL をリライトします。
6. ルールセットの名前を変更する場合、`<RuleSet id="ruleset_template">` 行を編集し、名前をルールセットの名前に置き換えます。
7. 「保存」をクリックします。

ルールセットをダウンロードする

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

ルールセットをダウンロードし、ファイルに保存することができます。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
2. ロケーションパネルで「サービス設定」を選択します。

3. ナビゲーションパネルの「リライター」の隣にあるプロパティの矢印をクリックします。
現在定義されている属性のリストが、データパネルに表示されます。
4. ファイルに保存するルールセットの「ダウンロード」リンクをクリックします。
5. ファイル名を指定し、保存します。

ルールセットをアップロードする

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

ルールセットのファイルをシステムにアップロードできます。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「リライター」の隣にあるプロパティの矢印をクリックします。
現在定義されている属性のリストが、データパネルに表示されます。
4. リストされるルールセットの隣にある「アップロード」リンクをクリックします。
5. 新規アップロードするルールセットのファイル名を参照して指定するか、直接入力します。
6. 「アップロード」をクリックします。

ファイル内の `<RuleSet id="ruleset_template">` 行で定義された名前がシステムのルールセット名と一致する場合、そのルールセットファイルはファイルの内容で置き換えられます。`<RuleSet id="ruleset_template">` 行で定義される名前が固有の場合、新しいルールセットがその名前で作成され、リストに追加されます。

既存のルールセットを削除する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「リライト」の隣にあるプロパティの矢印をクリックします。
現在定義されている属性のリストが、データパネルに表示されます。
4. 削除するルールセットの隣にあるチェックボックスをクリックします。
複数のルールセットを選択できます。
5. 「削除」をクリックします。
確認のメッセージが表示されます。
6. 「はい」をクリックして選択したルールセットを削除します。

デフォルトのルールセットを復元する

誤ってデフォルトのルールセットを削除した場合、次のようにして復元できます。

```
rwadmin store --runasdn "uid=amadmin, ou=people, o=sesta.com, o=isp"
--password "testing123" /resources/DefaultRuleSet.xml
```

ここで、"/resources/DefaultRuleSet.xml" は `rewriter.jar` ファイルに保存されたルールセットの場所です。

注 インストール時にパッケージされたデフォルトのルールセットが復元されます。デフォルトのルールセットをカスタマイズした場合、変更は復元されません。

検索エンジンサービスの管理

この章では、Sun Java™ System Portal Server の検索エンジンサービスの設定および管理方法について説明します。

この章で説明する内容は次のとおりです。

- [検索エンジンサービスの概要](#)
- [検索チャンネルの設定](#)
- [検索エンジンの管理](#)
- [ロボットの管理](#)
- [データベースの管理](#)
- [データベース分類の管理](#)

検索エンジンサービスの概要

Portal Server の検索エンジンは、Google や Alta Vista などの一般的なインターネット検索エンジンと同様のインタフェースで検索とブラウズをサポートする、分類およびデータベースサービスです。また、検索エンジンには、ドキュメントリソースを検索、変換、および要約するロボット型検索の機能もあります。Portal Server 6 では、デスクトップのみが JSP™ プロバイダを使用してインタフェースを提供しています。また、検索エンジンには、設定編集のための管理ツールおよびシステム管理のためのコマンド行ツールもあります。設定は、Sun Java™ System Access Manager のサービス属性値として Sun Java System Access Manager 管理コンソールで定義し、格納できます。

注 管理コンソールを使用すると、管理者は検索エンジンオプションの大部分を設定できますが、コマンドラインから使用できるすべての管理機能を実行できるわけではありません。

検索データベース

特定のリソースや複数の種類にまたがるリソースの検索には、データベースを使用します。データベースの個々のエントリをリソース記述 (RD) と呼びます。リソース記述とは、単一のリソースに関する特定の情報セットのことです。なお、各リソース記述のフィールドは、データベーススキーマによって設定されます。

データベースに RD を指定する場合、2つの方法があります。

- RD の作成 : 通常はこの方法を使用します。ロボットプロセスを使用してリソースを検索し、その記述を生成します。
- RD の交換 : 大規模な分散ネットワークインデックスに適しています。リモートシステムで生成した RD を、検索エンジンでデータベースにインポートします。

Portal Server 検索エンジンの RD は、SOIF (Summary Object Interchange Format) や RDM (Resource Description Message) などのオープンなインターネット標準に基づいています。このため、検索エンジンはクロスプラットフォームの企業環境でも動作します。

検索ロボット

データベースに情報を格納する方法の一つにロボットの使用が挙げられます。検索エンジンはロボットを使用してドメインのリソースを検索し、その結果を報告します。ロボットは小規模なプログラムですが、次の2つの機能を備えています。

- リソースへのリンクを抽出し、追跡する (列挙または巡回とも呼ばれる)
- RD を作成し、データベースに格納する (生成またはインデックス作成とも呼ばれる)

システム管理者として、さまざまな方法で、あらゆる局面にわたってこれらのプロセスを制御することができます。主な制御対象は次のとおりです。

- ロボットの開始、停止、およびスケジュールを設定して、ロボットの稼動時間を制御する。
- ロボットがアクセスするサイトを定義して、ロボットがリソースを検索する場所を制御する。
- 巡回属性を定義して、検索の頻度を制御する。
- フィルタを定義して、ロボットがインデックスを作成するリソースタイプを制御する。
- インデックス属性を定義して、データベースに作成するエントリの種類を制御する。

検索エンジンには、ロボットの実行結果を確認するためのユーティリティもあります。

データベース分類カテゴリ

ユーザーが検索システムと対話する場合、2つの方法があります。クエリーを直接入力してデータベースを検索する方法と、カスタムなカテゴリセットを使用してデータベースコンテンツ経由でブラウズする方法です。カテゴリの階層構造は分類と呼ばれます。リソースの分類は、データベースに目次を作成することに例えることができます。

ブラウズは、検索システムではオプションの機能です。つまり、カテゴリによるブラウズの機能がなくても、検索システムは十分有用なものになります。表示可能なカテゴリを追加することがインデックスのユーザーに役立つかどうか、さらに、作成するカテゴリの種類も検討する必要があります。

検索データベースのリソースは、検索手順を単純にするためにカテゴリに割り当てられます。データベースに多数の項目がある場合には、関連する項目をグループ化すると便利です。これによって、ユーザーは特定の種類の項目をすばやく検索したり、類似する項目と比較したり、目的の項目を選択したりすることができます。

このような分類は製品およびサービスのインデックスに共通です。衣料品のカタログでは、衣類を男性用、女性用、および子供用に分類して、さらに、これらのカテゴリをコート、シャツ、靴などに分類します。オフィス製品カタログの場合、家具は文具、コンピュータ、およびソフトウェアと区別されます。広告ディレクトリは、製品とサービスのカテゴリによって編成されます。

印刷物の索引で項目をカテゴリ別に分類するときの原則は、オンラインインデックスにも当てはまります。この発想では、種類別に検索することが容易になるので、ユーザーは目的のリソースを選択できます。設計するインデックスの適用範囲を問わず、カテゴリを設定する際にもっとも重要なのは使いやすさです。したがって、ユーザーがどのようにカテゴリを使用するかを知る必要があります。たとえば、異なる場所に3つの支社を持つ企業のためにインデックスを設計する場合、トップレベルのカテゴリを支社ごとに作成しようとするかもしれません。しかし、ユーザーにとって、支社の所在地よりも各部門の機能の方が重要であれば、リソースを部門別に分類することが適切です。

カテゴリを定義すると、カテゴリにリソースを割り当てるためのルールを設定する必要があります。このルールを、分類ルールと呼びます。分類ルールを適切に定義しないと、ユーザーがカテゴリでブラウズしようとしても、リソースを検索できないことがあります。リソースは正しく分類する必要がありますが、同時にドキュメントの分類も適切でなければなりません。

ドキュメントは、この値に設定した最大数を上限として、任意の数のカテゴリに割り当てることができます。分類ルールはフロー制御による判定を実行しないので、フィルタルールよりも単純です。分類ルールでは、特定のカテゴリをリソース記述の一部としてリソースに割り当てるための条件を決定する必要があります。分類ルールは、「もし<ある条件>が真ならば、リソースを<カテゴリ>に割り当てる」という形式の簡単な条件文で作成されます。

検索チャンネルの設定

ここでは検索エンジンサービスの初期設定の方法について説明します。設定は、Sun Java System Access Manager 管理コンソールを介して、Sun Java System Access Manager のサービス属性値として定義、格納できます。

検索サービスはグローバルに追加されるので、その設定は Portal Server 全体に適用されます。デフォルトでは、Portal Server のインストール時に指定した組織に、検索サービスが追加されます。サンプルのポータルをインストールすると、サンプルのポータルデスクトップの検索タブに検索チャンネルが配置されます。この検索チャンネルは Portal Server のインストール中に設定するものです。ただし、新規組織と新規インスタンスには、検索 URL を定義する必要があります。

検索プロバイダユーザーのデフォルトの動作では、ユーザーがクエリーを入力すると、「一致がありません」というメッセージが表示されます。

検索結果を取得するには、検索サーバーを設定してドキュメントデータベースを作成する必要があります。

検索サーバーの初期設定

次の手順にしたがって検索プロバイダを設定します。この手順はデータベースに取り込む方法の一例です。インポート機能を使用することもできます。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. 新規サイトを作成します。
 - a. 「ロボット」をクリックします。
 - b. 「サイト」をクリックします。
 - c. 「サイトの管理」の「新規」をクリックして、ロボットでインデックスを作成するサイトを定義します。

- d. サイト (URL またはドメイン) のタイプ、インデックスを作成するサイト、およびロボットが巡回する深さを指定します。
- e. 「サイトの作成」を選択してデフォルトの検索属性を使用します。あるいは、「サイトの作成と編集」を選択して検索サイトを詳細に定義します。

サイトを定義する検索属性の詳細については、『Portal Server Technical Reference Guide』を参照してください。

5. 分類を作成します。
「カテゴリ」の下で「カテゴリエディタ」を使用して分類を作成するか、サンプルの分類 SOIF ファイルを config/taxonomy.rdm にコピーします。
6. 使用しないデフォルトのフィルタを無効にします。
「ロボット」をクリックし、「フィルタ」をクリックします。「フィルタルール」リストで使用しないフィルタをオフにします。
7. (省略可能) カテゴリ別にドキュメント結果を取得する必要があるなら、ここでロボット分類ルールを作成します。
「カテゴリ」の下にある「分類ルールエディタ」を使用して、ロボット分類ルールを作成できます。
8. ロボットを開始します。
「ロボット」、「概要」をクリックし、次に「開始」をクリックし、ロボットを開始します。
9. カテゴリのインデックスを再作成します。
「カテゴリ」、「インデックスの再作成」をクリックして、インデックスを再度作成します。

検索 URL を定義する

searchServer プロパティは検索 URL を定義します。デフォルトの組織の場合、この値は自動的に設定されます。ただし、新規に組織や SearchProvider インスタンスを作成する場合、またはサンプルの dp-org.xml を手動でロードする場合には定義されません。値が定義されていない場合にユーザーが検索を実行すると、ユーザーのデスクトップに次のエラーメッセージが表示されます。

```
You got a
com.sun.portal.search.providers.taglib.SearchTaglibException:Search
Request Error:search server is not defined.
```

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. 検索 URL を定義する組織を選択します。
3. 「表示」メニューから「サービス」を選びます。
ナビゲーションパネルの「表示」メニューとロケーションパネルの「場所」パスを使用します。
4. ナビゲーションパネルの「ポータルデスクトップ」の隣にあるプロパティ矢印を選択します。
デスクトップ属性ページがデータパネルに表示されます。
5. デスクトップページで「チャンネルおよびコンテナを管理」リンクを選択します。
「チャンネル」ページが表示されます。ページの上部にコンテナのパスが表示されます。定義済みのチャンネルがリストに表示されます。
6. 変更する検索チャンネルの隣の「プロパティの編集」リンクを選択します。
「プロパティ」ページが表示されます。
7. 次の書式で検索サーバープロパティに検索 URL を指定します。
http://portal_server_name:port/portal/search
8. 「保存」をクリックします。
9. 検索 URL を確認する場合、次の操作を行います。

- a. 検索 URL を設定した組織にログインします。たとえば、次のように B という名前の組織にログインします。

`http://portal_server_name:port/amserver/ui/login?org=B`

- b. 検索チャンネルから検索を実行します。

検索エンジンの管理

検索エンジンの初期設定とデータベースの生成が完了すると、Sun Java System Access Manager 管理コンソールで、検索エンジンの表示と管理ができるようになります。

検索エンジン操作の表示、管理、監視

検索エンジンの操作属性には、基本と詳細の 2 つのレベルがあります。基本設定ページは、検索サービスを管理コンソールから選択したときにデフォルトで表示されます。表示される基本設定には、サーバールート、一時ファイルの場所、およびドキュメントレベルのセキュリティが含まれます。詳細設定には、検索エンジンの各種コンポーネントのログの場所と、設定されたログレベルが含まれます。

さらに、管理コンソールを使用して、管理者はログファイルから抽出されたログファイルまたは特定の情報を表示することができます。

基本設定を表示または管理する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーで「サーバー」、「設定」の順に選択します。
5. 検索エンジンのサーバールートのディレクトリを表示します。

6. 検索エンジンの一時ファイルのディレクトリを表示または指定します。
7. ドキュメントレベルのセキュリティ属性を表示または指定します。
「オフ」の場合、すべてのユーザーがこのデータベースの RD にアクセスできます。「オン」の場合、RD の ReadACL フィールドを評価して、ユーザーに RD へのアクセス権があるかどうかを判断する必要があります。
8. 「保存」を選択して、変更した属性を記録します。

詳細設定を表示または管理する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーで「サーバー」、「詳細」の順に選択します。
5. 詳細属性を表示または指定します。
ここでは、「検索 (rdm)」、「検索ログを無効にする」、「インデックスのメンテナンス」、「RD マネージャ」、「RDM サーバー」、および「ログレベル」が設定できません。
6. 「保存」を選択して、変更した設定を記録します。

検索エンジンの状況を監視する

検索エンジンは、検索状況を監視するための多数のレポートを提供します。

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

各種レポートを表示するには、次の手順に従います。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. 「レポート」を選択します。
5. メニューバーのリンクを選択すると特定のレポートが表示されます。
ここでは、次のレポートオプションを使用できます。「開始位置」、「除外 URL」、「ロボットの詳細レポート」、「ログファイル」、「頻度の高い検索」です。

ロボットの管理

ロボットを管理するために必要な設定とメンテナンスのタスクを次に示します。

- [サイトの定義](#)
- [ロボットの巡回の制御](#)
- [ロボットデータのフィルタリング](#)
- [インデックス作成の属性の定義](#)
- [ロボットユーティリティの使用](#)
- [ロボットのスケジュール設定](#)

サイトの定義

ロボットはリソースを見つけて、それらの記述をデータベースに追加するかどうか（およびその方法）を決定します。どのサーバーを訪問し、サーバーのどの部分のインデックスを作成するかの判断が、サイト定義に呼び出されます。

検索エンジンのサイトの定義は、サーバー管理者の最も重要な作業の1つです。ロボットは、インデックス作成が必要なすべてのサーバーに送信しなければなりません。同時に、データベース領域を過度に拡張して正確な情報の検索を困難にする可能性のある外部サイトを排除することも必要です。

ロボットでインデックスを作成するサイトを定義する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーで「ロボット」、「サイト」の順に選択します。
5. 次のようにサイトを作成します。
 - a. 「新規」をクリックします。
 - b. サイトのタイプ (URL またはドメイン) を選択します。
 - c. サイトと深さを指定します。
 - d. 「保存」をクリックします。
6. サイト属性を編集するには、「編集」リンクをクリックします。
サイト属性の形式が表示されます。検索サイトについての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。
 - e. 属性を編集します。
 - f. 「保存」をクリックします。

ロボットの巡回の制御

ロボットはインデックスの作成のために選択したさまざまなサイトにクローリングします。管理者は巡回操作パラメータを定義することで、ロボットのサイト検索方式を制御できます。巡回パラメータにより、速度、完了動作、ログインレベル、規格への準拠、認証パラメータ、プロキシ設定、追従する最大リンク数、その他の設定を定義できます。ロボットの巡回についての説明は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

ロボットの巡回を制御する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーで「ロボット」を選択し、次に「巡回」を選択します。
これにより、ロボットの巡回操作パラメータと設定を定義する属性のフォームが表示されます。ロボットの巡回属性についての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。
5. 必要に応じてロボットの巡回属性を変更します。

注 server.xml ファイルにプロキシ設定 (http.proxyHost= と http.proxyPort= オプションを使用) がある場合は、「すべてのホストからのコマンドを受け入れる」にチェックマークを付けてからロボットを実行する必要があります。

6. 「保存」をクリックします。

ロボットデータのフィルタリング

フィルタを使用すると、リソースの属性をフィルタ定義と比較することでリソースを識別できるようになるので、これをサイト定義から除外することも、サイト定義に含むこともできます。ロボットには多くの事前定義されたフィルタがあります。一部のフィルタはデフォルトで有効です。次のフィルタは事前定義されています。アスタリスク (*) の付いたファイルがデフォルトで有効です。

- アーカイブファイル *
- オーディオファイル *
- バックアップファイル *
- バイナリファイル *
- CGI ファイル *
- 画像ファイル *
- Java、JavaScript、スタイルシートファイル *
- ログファイル *
- Power Point ファイル
- リビジョン管理ファイル *
- ソースコードファイル *
- 一時ファイル *
- ビデオファイル *
- スプレッドシートファイル
- プラグインファイル
- Lotus Domino ドキュメント
- Lotus Domino オープンビュー
- システムディレクトリ (UNIX)
- システムディレクトリ (NT)

フィルタリングプロセスの管理には、新規フィルタ定義を作成する方法、フィルタ定義を変更する方法、およびフィルタを有効、または無効にする方法があります。

新規フィルタ定義を作成する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーから「ロボット」、「フィルタ」の順に選択します。
5. 「新規」を選択して、新規フィルタのニックネームを指定します。
6. 「フィルタ定義」では、チェックボックスにチェックマークを付けて、「フィルタソース」、「選別基準」、および「フィルタ文字列」の値を指定します。必要に応じて、任意の数の「フィルタ定義」を指定することができます。
7. フィルタの説明を入力します。
8. 新規サイトの作成時にこのフィルタを使用するには「新規サイト」にチェックマークを付けます。
9. このフィルタに一致するリソースを含めるか除外するかをボタンで指定します。
10. 「保存」をクリックします。

既存のフィルタ定義を変更する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。

2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーから「ロボット」、「フィルタ」の順に選択します。
5. 「フィルタルール」リストから変更するフィルタを検索し、「編集」リンクをクリックします。
6. 必要に応じてフィルタを変更します。
7. フィルタの説明を入力します。
8. 「保存」をクリックします。

フィルタを有効または無効にする

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーから「ロボット」、「フィルタ」の順に選択します。
5. 「フィルタルール」リストで変更するフィルタを検索します。
6. フィルタのオンまたはオフを示すボタンを選択します。
7. 「保存」をクリックします。

インデックス作成の属性の定義

ロボットフィルタを通過する各リソースに、ロボットによってデータベースに格納する RD を生成します。RD 生成を設定するときの選択によって、データベース検索時にユーザーに表示する範囲が決定されます。たとえば、ドキュメントの全文や、ドキュメントの最初の固定部分だけに対してインデックスを作成することができます。

インデックス作成の属性を定義する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーで「ロボット」、「インデックス作成」の順に選択します。
ロボットによるインデックス作成操作パラメータと設定を定義する属性のページが表示されます。ロボットのインデックス作成属性についての詳細は、『Portal Server Technical Reference Guide』を参照してください。
5. 必要に応じてロボットのインデックス作成の属性を変更します。
6. 「保存」をクリックします。

ロボットユーティリティの使用

ロボットには次の2つのデバッグツールまたはユーティリティがあります。

- **サイトプローブ**: DNS エイリアス、サーバーリダイレクト、仮想サーバーなどを確認します。
- **シミュレータ**: 特定の URL でロボットフィルタリングを部分的にシミュレートします。確認する URL を入力して「了解」を選択します。シミュレータは表示されたサイトをロボットが受け入れるかどうかを示します。

サイトプローブユーティリティを実行する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーで「ロボット」、「サイトプローブ」の順に選択します。
5. プローブするサイトの URL を入力します。
6. プローブで DNS 情報を取得するには、「DNS 情報の詳細を表示」を選択します。
7. 「了解」を選択して、サイトプローブを開始します。

シミュレータを実行する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーで「ロボット」、「シミュレータ」の順に選択します。
5. シミュレーションを実行する URL を入力します。
6. シミュレータでエイリアスを確認するには、「DNS エイリアスを確認」を選択します。

7. シミュレータでリダイレクトを確認するには、「サーバーリダイレクトを確認 (302)」を選択します。
8. 「了解」を選択して、シミュレータを開始します。

ロボットのスケジュール設定

検索データを最新に維持するには、ロボットはサイトを定期的に検索して、インデックスを作成する必要があります。ロボットの巡回とインデックス作成は処理リソースとネットワークの帯域幅を消費する場合があります。これらのリソースの制約を避けるために、ピーク以外の時間帯にロボットを実行するようにスケジュール設定する必要があります。管理コンソールを使用すると、管理者はロボットを実行する日時を指定して cron ジョブをセットアップすることができます。

ロボットのスケジュールを設定する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーで「ロボット」、「スケジュール」の順に選択します。
5. ロボットを開始する曜日と時刻 (時間と分) を選択します。
6. ロボットを停止する曜日と時刻を選択します。
7. 「保存」をクリックします。

データベースの管理

検索エンジンは RD をデータベースに格納します。次にデータベースの管理に必要な設定と保守タスクを示します。

- データベースへのインポート
- リソース記述の編集
- データベーススキーマの編集
- スキーマのエイリアスの定義
- データベース分析の表示
- データベースのインデックスの再作成
- データベースの期限切れ
- データベースの破棄
- データベースのパーティションの作成

データベースへのインポート

通常、検索データベースの項目はロボットから取得します。ユーザーはロボットにアクセスするサイトを指定し、ロボットはそのサイトのすべてのリソースを検索して RD を作成します。ただし、他の Portal Server 検索エンジン、iPlanet Web Server や Netscape™ Enterprise Server、または別のソースから生成したデータベースから既存の項目のデータベースをインポートすることもできます。ロボットを使用して新規に作成する代わりに RD の既存データベースをインポートすると、ネットワークトラフィックが大幅に減少し、大量のインデックス作成作業を分割できるので、より高速にインデックスを作成できます。中央データベースがインデックスを作成するサーバーから物理的に遠くにある場合には、RD をローカルに生成して、中央データベースに各地点のリモートデータベースを定期的にインポートすると便利です。

検索エンジンは、インポートエージェントを使用して、別のサーバーやデータベースから RD をインポートします。インポートエージェントは、外部ソースから RD を検索して、ローカルデータベースに情報をマージするプロセスです。RD のインポート元、インポート元で検索するデータ、およびこのジョブの実行方法を詳細に調整するための情報を指定するパラメータがあります。

データベースをインポートする前に、インポートエージェントを作成する必要があります。エージェントを作成したら、すぐにインポートプロセスを開始することも、インポートプロセスのスケジュールを設定してから実行することもできます。

インポートエージェントを作成する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. 「データベース」を選択し、次に「エージェントのインポート」リンクをクリックします。
5. 「新規」をクリックします。
インポートエージェントの属性ページが表示されます。
6. インポートエージェントに適切な属性を指定します。
データベースのインポート属性についての詳細は、『Portal Server Technical Reference Guide』を参照してください。
 - a. ソースがローカルファイルであるか、検索サーバーであるかを指定します。
 - b. ソースがファイルの場合には、ローカルファイルパスを指定します。
 - c. ソースが別の検索サーバーの場合には、リモートサーバー、インスタンス名、および検索 URI の URL を指定します。
 - d. インポートするデータベースの名前を指定します。
 - e. インポートエージェントの文字セットを指定します。
7. 「保存」をクリックします。

既存のインポートエージェントを編集する

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。

デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. 「データベース」を選択し、次に「エージェントのインポート」リンクをクリックします。
5. 編集するエージェントの右側にある「編集」リンクをクリックします。
6. インポートエージェントに適切な属性を指定します。

データベースのインポート属性についての詳細は、『Portal Server Technical Reference Guide』を参照してください。
7. 「保存」をクリックします。

リソース記述の編集

リソース記述の内容の変更が必要になることがあります。たとえば、元のドキュメントからリソース記述にコピーしたときに、スペルミスまでコピーした場合には、スペルの修正が必要になります。

リソース記述を編集する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。

デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。

4. メニューバーで「データベース」を選択し、次に「リソース記述」を選択します。
5. 検索する RD のタイプを選択して編集します。
ここでは、「すべての RD」、「カテゴリ化されていない RD」、「カテゴリ化された RD」、「カテゴリごとの RD」、「URL 特定の RD」、「RD を検索」を選択できます。
6. 「RD を検索」では、リソース記述で検索するテキスト文字列を指定します。
7. 「検索」をクリックします。
8. 見つかったリソース記述のリストで、編集するリソース記述を選択します。
9. 適切なリソース記述属性を編集します。
10. 「保存」をクリックします。

データベーススキーマの編集

スキーマは、検索エンジンで管理するリソースごとの情報とその形式を決定します。スキーマの設計によって、インデックスの有用性に影響する次の 2 つの要因が決まります。

- ユーザーがリソースを検索する方法
- ユーザーがリソース情報を表示する方法

スキーマはデータベースのリソース記述のマスタデータ構造です。データ構造フィールドのインデックス作成方法と定義方法によって、リソースへのアクセス頻度が変わります。

スキーマは、検索エンジンとそのロボットが使用するファイルの構造に密接に結びついています。管理コンソールでスキーマツールを使用してデータ構造を変更するだけです。スキーマファイル (schema.rdm) は単なるテキストファイルですが、直接編集しないでください。

検索エンジンのデータベーススキーマを編集すると、新規のスキーマ属性の追加、スキーマ属性の編集、あるいは属性の削除ができます。

スキーマには次の属性があります。

- **編集可能** : オンの場合、この属性はリソース記述エディタに表示される属性を示すので、ユーザーは値を変更できます。リソース記述エディタについては、[304 ページの「リソース記述の編集」](#)を参照してください。
- **インデックス作成可能** : この属性は「詳細検索」画面のポップアップメニューにフィールドが表示されることを示します。ユーザーはこの特定のフィールド内の値を検索できます。
- **説明** : これはスキーマの記述に使用するテキスト文字列です。コメントや注釈に使用します。

- エイリアス: この属性は、インポートしたデータベーススキーマ名をユーザー独自のスキーマに変換するためのエイリアスを定義します。

データベーススキーマを編集する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーで「データベース」を選択し、次に「スキーマ」を選択します。「スキーマ」属性ページが表示されます。
5. スキーマに新しい属性を追加するには、次の手順を実行します
 - a. 「スキーマリスト」の下の「新規」を選択します。
 - b. 「名前」および「説明」フィールドに、新しい属性の名前と説明を入力します。
 - c. 「編集可能」にチェックマークを付けて、属性を編集可能にします。
 - d. 「インデックス作成可能」にチェックマークを付けて、属性インデックスを作成可能にします。
6. 既存の属性を編集可能にするか、あるいはインデックスを作成可能にするには、次の手順を実行します。
 - a. スキーマのリストから属性の隣にある「編集」リンクをクリックします。「スキーマ」属性ページが表示されます。
 - b. 「編集可能」にチェックマークを付けて、属性を編集可能にします。
 - c. 「インデックス作成可能」にチェックマークを付けて、属性インデックスを作成可能にします。
 - d. 「更新」をクリックします。
7. 属性を削除するには、次の手順を実行します。

- a. スキーマリストで属性にチェックマークを付けます。
- b. 「削除」をクリックします。

注 検索エンジンのスキーマを変更するには、データベース全体のインデックス作成を再実行し、サーバーを再起動しなければならないことがあります。これは、検索エンジンのハイライト機能がスキーマフィールドの順序とタイプに影響されるためです。テキストフィールドの追加と削除（削除後に再追加した場合も同様）を行うと、検索結果のハイライト表示が不正確になる可能性が高くなります。

スキーマのエイリアスの定義

一部のインスタンスの名前は、データベーススキーマのフィールド名と矛盾している可能性があります。1つはサーバーから別のサーバーへリソース記述をインポートするときのものです。2つのサーバーのスキーマの項目名が同一であることを、常に保証することはできません。同様に、ロボットが HTML META タグをドキュメントからスキーマフィールドに変換するときは、名前はドキュメントによって制御されます。

検索エンジンは、ユーザーのスキーマ属性にスキーマのエイリアスを定義して、外部スキーマ名をデータベースのフィールドに有効な名前にマッピングできるようにします。

スキーマのエイリアスを定義する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. 「データベース」を選択し、次に「スキーマ」リンクをクリックします。
「スキーマ」属性ページが表示されます。

5. エイリアスを定義する属性をクリックします。
6. インポートしたデータベースで使用するフィールド名を指定します。
7. 「更新」をクリックします。
8. 「インデックスの再作成」をクリックします。

大規模なデータベースの場合、インデックスの再作成プロセスに数時間かかることがあります。

データベース分析の表示

検索エンジンのレポートには、インデックスを作成したサイトの数、およびデータベースの各サイトのリソースの数を含む情報が含まれます。

データベース分析情報を表示する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーで「データベース」、「解析」の順に選択します。
現在検索データベースに存在するすべてのサイトをソートしたリストと、各サイトのリソースの数です。
5. 「保存」を選択して最新のリストを生成します。

データベースのインデックスの再作成

一部のインスタンスでは、検索エンジンが使用するリソース記述データベースのインデックスを再作成する必要があります。わかりやすい例は、スキーマを編集してインデックスフィールドを追加または削除する場合です。

ディスクエラーによってインデックスファイルが破損した場合にも、データベースのインデックスを再作成する必要があります。また、新規のリソース記述を大量に追加したあとにインデックスを再作成することも有効です。

データベースのインデックスを再作成するのに、数時間を要する場合があります。

データベースのインデックスの再作成に要する時間は、データベースに含まれるレコードの数に比例します。大規模なデータベースでは、サーバーの負荷が低い時にインデックスの再作成を行うようにしてください。

データベースのインデックスを再作成する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. 「データベース」を選択し、次に「管理」を選択します。
5. 「データベースリスト」の下の「インデックスの再作成」をクリックします。
6. 「データベースインデックスの再作成」にチェックマークを付けて、「了解」をクリックします。
検索エンジンによって、検索コレクションとそのインデックスファイルが再構築されます。

データベースの期限切れ

データベースが期限切れになると、古くなったリソース記述も期限切れになります。リソース記述が期限切れになるのは、期限切れを実行するときだけです。期限切れになったリソース記述が削除されても、データベースのサイズは減少しません。

データベースを期限切れにする

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. 「データベース」を選択し、次に「管理」を選択します。
5. 「データベースリスト」の下の「期限切れ」を選択します。
6. 期限切れにするデータベースを確認し、「了解」をクリックします。

データベースの破棄

リソース記述の属性の1つに有効期限があります。ロボットは HTML META タグ、またはリソースのサーバーが提供する情報を基に有効期限を設定します。リソースが有効期限を指定しない限り、リソース記述はデフォルトで作成から3カ月有効です。期限切れのリソース記述は、検索エンジンによって定期的にデータベースから破棄される必要があります。

破棄を実行すると、データベースの中身を削除できます。これによって、インデックス用のディスク領域は回復しますが、メインデータベースのディスク領域は回復できません。これは、この領域が新規のデータをデータベースに追加したときに再利用されるためです。

サーバーから期限切れのリソース記述を破棄する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. 「データベース」を選択し、次に「管理」を選択します。
5. 「データベースリスト」の下の「破棄」を選択します。
6. 「データベースの破棄」にチェックマークを付けて、「了解」をクリックします。
破棄が完了すると、システムに破棄が正常に終了したことを示すメッセージが表示されます。

データベースのパーティションの作成

検索エンジンを使用すれば、複数のディスク、ファイルシステム、ディレクトリ、またはパーティションにまたがる検索データベースの物理ファイルを分割できます。異なる物理デバイス、または論理デバイスにデータベースを拡張することで、単一のデバイスを使用するよりも大きなデータベースにできます。

デフォルトでは、検索エンジンは1つのディレクトリだけを使用するデータベースを設定します。コマンドラインインタフェースを使用すると、データベースパーティションに次の2種類の操作を実行できます。

- 新しいパーティションの追加
- パーティションの移動

検索エンジンは、各パーティションに空き領域があるかどうかを確認しません。データベースに適切な空き領域を確保するのは、ユーザーの作業です。

データベースのパーティションは、合計で最大 15 個追加できます。

注 一度パーティションの数を増やすと、あとでパーティションの数を減らすには、データベース全体を削除する必要がありますので注意してください。

新しい場所の名前を指定することで、データベースパーティションの物理的な場所も変更できます。同様に、既存のパーティションの名前も変更できます。パーティションを操作するには `rdmgr` コマンドを使用します。`rdmgr` コマンドについての詳細は、『Sun Java System Portal Server 2005Q1 Technical Reference Guide』を参照してください。

データベース分類の管理

次に、データベース分類を管理するために必要な設定とメンテナンスのタスクを示します。

- [カテゴリの設定](#)
- [分類ルールの定義](#)

カテゴリの設定

Sun Java System Access Manager 管理コンソールを使用してデータベース分類を設定するには、次の手順を実行します。

- [サブカテゴリを作成する](#) (親カテゴリのサブカテゴリ)
- [カテゴリを更新する](#)
- [カテゴリを削除する](#)

サブカテゴリを作成する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。

4. メニューバーから「カテゴリ」、「カテゴリエディタ」の順に選択します。
5. サブカテゴリを作成するカテゴリを選択します。
それまでにカテゴリを定義していない場合、一覧表示されるのは、「検索」というルートカテゴリだけです。下位への検索リンクを選択して、ルートカテゴリを展開します。
6. 「名前」フィールドに、カテゴリの名前を指定します。
7. 「説明」フィールドに、カテゴリの説明を指定します (省略可能)。
8. 「サブカテゴリとして追加」をクリックして、カテゴリを作成します。
9. 「保存」をクリックします。

注 表示可能なカテゴリが複数ページにまたがる場合、カテゴリエディタには移動先のリストが表示されます。ページを上下に移動するボタンを使用して、現在のページから1ページずつ上下にスクロールします。複数のページを移動するには、移動先ボタンを使用します。

カテゴリを更新する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーから「カテゴリ」、「カテゴリエディタ」の順に選択します。
5. 更新するカテゴリを選択します。
6. カテゴリ名を変更するには、「名前」フィールドにカテゴリの新しい名前を指定します。
7. カテゴリの説明を変更するには、「説明」フィールドにカテゴリの説明を指定します。
8. 「更新」をクリックします。

9. 「保存」をクリックします。

注 表示可能なカテゴリが複数ページにまたがる場合、カテゴリエディタには移動先のリストが表示されます。ページを上下に移動するボタンを使用して、現在のページから1ページずつ上下にスクロールします。複数のページを移動するには、移動先ボタンを使用します。

カテゴリを削除する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーから「カテゴリ」、「カテゴリエディタ」の順に選択します。
5. 削除するカテゴリを選択します。
カテゴリを削除すると、すべてのサブカテゴリも削除されます。
6. 「削除」をクリックします。
7. 「保存」をクリックします。

注 表示可能なカテゴリが複数ページにまたがる場合、カテゴリエディタには移動先のリストが表示されます。ページを上下に移動するボタンを使用して、現在のページから1ページずつ上下にスクロールします。複数のページを移動するには、移動先ボタンを使用します。

分類ルールの定義

分類ルールは簡単な条件文です。その形式は「<ある条件> が真であれば、リソースを<カテゴリ>に割り当てる」です。

分類ルールを定義する

注 Access Manager 管理コンソールに関する最新の詳細情報については、『Sun Java System Access Manager 2005Q1 管理ガイド』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. ナビゲーションパネルの「検索」の隣にあるプロパティの矢印を選択します。
4. メニューバーで「カテゴリ」、「分類ルールエディタ」の順に選択します。
5. 新しいルールを作成する場合は、「新規」をクリックします。
6. 既存のルールを編集する場合は、そのルールを選択します。
7. ドロップダウンメニューから、使用する要素タイプまたは属性をクリックしてリソースを分類します。
8. ドロップダウンメニューから、比較テストを選択します。
使用可能な比較テストは「次と等しい場合」、「次を含む場合」、「次から始まる場合」、「次で終わる場合」、または正規表現です。
9. 比較するテキスト文字列を定義します。
10. 比較が真の場合にリソースを分類するカテゴリをクリックします。
11. 「保存」をクリックします。

検索エンジンロボットの管理

Sun Java™ System Portal Server 検索エンジンロボットと、それに対応する設定ファイルについて説明します。この章で説明する内容は次のとおりです。

- 検索エンジンロボットの概要
- ロボットプロセスパラメータの設定
- フィルタリング処理
- ユーザーが変更できるパラメータ
- `robot.conf` ファイルの例

検索エンジンロボットの概要

検索エンジンロボットは、ドメイン内のリソースを特定し、レポートを作成するエージェントです。これには、列挙子フィルタとジェネレータフィルタの 2 種類のフィルタを使用します。

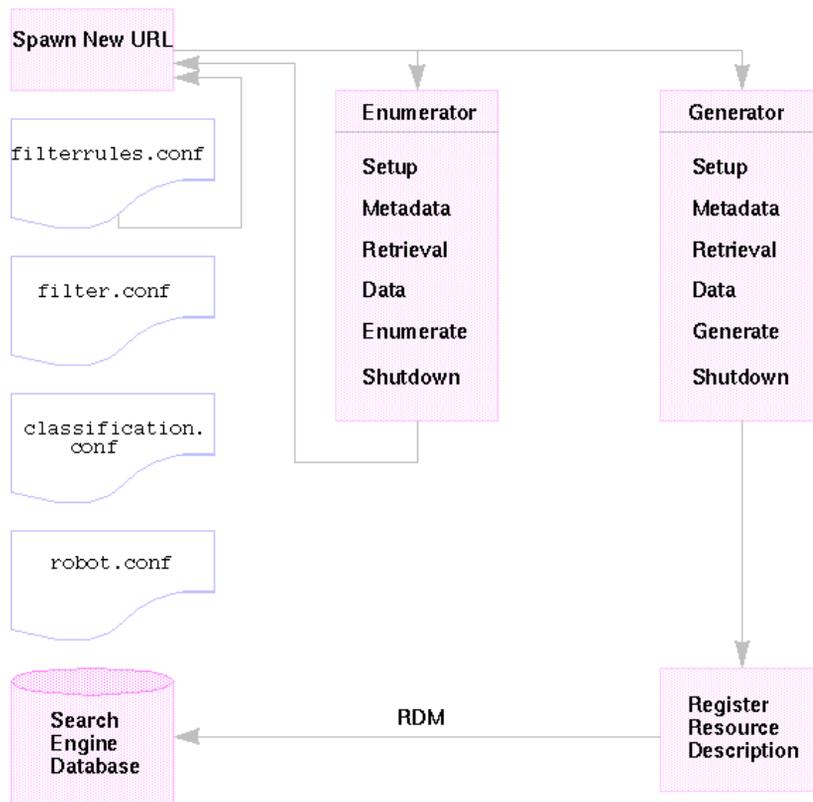
列挙子フィルタは、ネットワークプロトコルを使用してリソースを検出します。列挙子フィルタは、各リソースをテストし、適切な基準に一致した場合はリソースを列挙します。たとえば、列挙子フィルタは HTML ファイルからハイパーテキストリンクを抽出し、そのリンクを使用して別のリソースを検索できます。

ジェネレータフィルタは各リソースをテストし、リソース記述 (RD) を作成する必要があるかどうかを確認します。リソースがテストに合格した場合、ジェネレータは検索エンジンデータベースに格納される RD を作成します。

ロボットの動作の仕組み

図 14-1 は、検索エンジンロボットの動作の仕組みを示しています。図 14-1 に示されるように、ロボットは URL および関連のあるネットワークリソースを調べます。各リソースは、列挙機能と生成機能の両方によってテストされます。リソースが列挙テストに合格すると、ロボットがそのリソースをほかの URL について検査します。リソースがジェネレータテストに合格すると、ロボットは検索エンジンデータベースに格納されているリソース記述を生成します。

図 14-1 ロボットの動作の仕組み



ロボット設定ファイル

ロボット設定ファイルでは、検索エンジンロボットの動作が定義されます。これらのファイルは、`/var/opt/SUNWps/http-hostname-domain/portal/config` ディレクトリにあります。表 14-1 は、各ロボット設定ファイルを説明しています。この表には 2 つの列があります。最初の列は設定ファイル、2 番目の列はファイルの内容の説明を示します。

表 14-1 ロボット設定ファイル

ロボット設定ファイル	説明
<code>classification.conf</code>	ロボットが生成する RD の分類に使用されるルールを含みます。
<code>filter.conf</code>	検索エンジンロボットで列挙および生成の際に利用できるすべてのフィルタを含みます。列挙フィルタ、生成フィルタのいずれにも同じフィルタルールが含まれるので、両タイプのフィルタに共通のルール変更ができます。 参照により、このファイルには、 <code>filterrules.conf</code> に格納されるフィルタリングルールも含まれます。
<code>filterrules.conf</code>	開始ポイント (シード URL と呼ばれる) およびフィルタリングルールを含みます。
<code>robot.conf</code>	ロボットのオペレーティングパラメータのほとんどが定義されます。さらに、このファイルは <code>filter.conf</code> ファイル中の使用すべき適切なフィルタをロボットに指示します。

注 検索サービスは、この他に `convert.conf` と `import.conf` の 2 つの設定ファイルを使用します。これらのファイルは検索サーバーにより生成され、一般に手動では編集できません。

検索エンジン管理インターフェースを使用すれば、ほとんどのパラメータを設定できるので、通常は `robot.conf` ファイルを編集する必要はありません。

ただし、上級ユーザーはこのファイルを手動で編集することにより、インターフェースを介してでは設定できないパラメータを設定することができます。

ロボットプロセスパラメータの設定

robot.conf ファイルは、filter.conf にある適切なフィルタにロボットを指定するなど、ロボット用の多くのオプションを定義します (旧バージョンとの下位互換性のために、robot.conf にシード URL を含めることもできる)。

Sun Java System Access Manager 管理コンソールは、robot.conf ファイルを編集するために使用されます。手動で編集できるパラメータについては、[324 ページの「ユーザーが変更できるパラメータ」](#)で詳細に説明します。

もっとも重要なパラメータは、enumeration-filter および generation-filter で、これらのパラメータは、ロボットが列挙および生成で使用するフィルタを指定します。これらのパラメータのデフォルト値は、enumeration-default および generation-default です。これらは、filter.conf ファイルにおいてデフォルトで提供されるフィルタ名です。

フィルタはすべて filter.conf ファイルで定義される必要があります。filter.conf で独自のフィルタを定義する場合、必要なパラメータを robot.conf に追加する必要があります。

たとえば、my-enumerator という名前の新しい列挙フィルタを定義する場合、robot.conf に次のパラメータを追加します。

```
enumeration-filter=my-enumerator
```

フィルタリング処理

ロボットはフィルタを使用して、処理するリソースとそのリソースの処理方法を決定します。ロボットがリソースと同時にリソースの参照を検出すると、各リソースにフィルタを適用して、それらのリソースを列挙し、検索エンジンデータベースに格納するリソース記述を生成するかどうかを決定します。

ロボットは1つあるいは複数のシード URL を調べ、フィルタを適用し、シード URL などを列挙して生成された URL にそのフィルタを適用します。シード URL は filterrules.conf ファイルで定義されます。

フィルタは、必須である初期化操作を実行し、現在のリソースに比較テストを適用します。各テストは、リソースを許可または拒否することが目標です。フィルタにはシャットダウンフェーズも備わっており、必要なクリーンアップ操作を実行します。

リソースが許可されるということは、フィルタ通過の続行が許可されることです。リソースが否認される場合、そのリソースは拒否されます。拒否されたリソースのフィルタは、それ以上動作しません。リソースが拒否されない場合、最終的にロボットはそのリソースを列挙し、さらにほかのリソースを検出しようとします。ジェネレータはリソース記述も作成できます。

これらの操作は必ずしも連携されていません。リソースには列挙につながるものもあれば、RD 生成につながるものもあります。多くのリソースは列挙にも、RD 生成にもつながります。たとえば、リソースが FTP ディレクトリである場合、一般的にはそのリソース用に RD は生成されません。ただし、ロボットは FTP ディレクトリの個別ファイルを列挙できます。ほかのドキュメントへのリンクを含む HTML ドキュメントは、RD を受信し、また、関連のドキュメントを列挙することもできます。

次の節では、フィルタ処理について詳しく説明します。

- [フィルタ処理の段階](#)
- [フィルタの構文](#)
- [フィルタディレクティブ](#)
- [フィルタの記述または修正](#)

フィルタ処理の段階

列挙フィルタ、生成フィルタのどちらにも、フィルタリング処理における 5 つのフェーズがあります。どちらのフィルタにも、共通する 4 つのフェーズがあります。

これらのフェーズは次のとおりです。

- **セットアップ**: 初期化処理を行います。ロボットの生存期間において一度だけ発生します。
- **メタデータ**: そのリソースについて利用可能なメタデータに基づき、リソースをフィルタリングします。リソースがネットワークを介して取得される前に、メタデータのフィルタリングがリソースごとに一回実行されます。表 14-2 は、共通のメタデータタイプの例を示しています。この表では 2 つの列があります。最初の列はメタデータのタイプ、2 番目の列は説明、3 番目の列は例を示します。

表 14-2 共通のメタデータタイプ

メタデータ	説明	例
完全な URL	リソースの場所	http://home.siroe.com/
プロトコル	URL のアクセス部分	http、ftp、file
ホスト	URL のアドレス部分	www.siroe.com
IP アドレス	ホストの数値バージョン	198.95.249.6
PATH	URL のパス部分	/index.html
深さ	シード URL からのリンク数	5

- データ: リソースのデータに基づいてリソースをフィルタリングします。データのフィルタリングは、ネットワークを介して取得されたあとにリソースごとに1回実行されます。フィルタリングに使用できるデータには、次のものがあります。
 - content-type
 - content-length
 - content-encoding
 - content-charset
 - last-modified
 - expires
- 列挙: 検査すべきほかのリソースを参照するかどうかを判断するため、現行のリソース中の参照を列挙します。
- 生成: リソースのリソース記述 (RD) を生成し、検索エンジンデータベースにそのRDを保存します。
- シャットダウン: 必要な終了操作を実行します。ロボットの生存期間において一度だけ発生します。

フィルタの構文

filter.conf ファイルには、列挙フィルタおよび生成フィルタの定義が含まれています。このファイルには、列挙と生成の両方について、複数のフィルタを含むことができます。robot.conf ファイルの enumeration-filter パラメータおよび generation-filter パラメータでフィルタを指定するため、使用するフィルタをロボットが判断できます。

フィルタの定義は、ヘッダー、本文、終了という明確に定義された構造になります。ヘッダーは、フィルタの開始を識別し、たとえば次のように名前を宣言します。

```
<Filter name="myFilter">
```

本文は、セットアップ、テスト、列挙または生成、およびシャットダウン中のフィルタの動作を定義する一連のフィルタディレクティブで構成されています。各ディレクティブでは、関数とその関数のパラメータ (該当する場合) を指定します。

終了は </Filter> によって示されます。

[323 ページのコード例 14-1](#) は、enumeration1 という名のフィルタを示しています。

コード例 14-1 列挙ファイルの構文

```
<Filter name="enumeration1">
  Setup fn=filterrules-setup config=./config/filterrules.conf
  # Process the rules
  MetaData fn=filterrules-process
  # Filter by type and process rules again
  Data fn=assign-source dst=type src=content-type
  Data fn=filterrules-process
  # Perform the enumeration on HTML only
  Enumerate enable=true fn=enumerate-urls max=1024 type=text/html
  # Cleanup
  Shutdown fn=filterrules-shutdown
</Filter>
```

フィルタディレクティブ

フィルタディレクティブは、ロボットアプリケーション関数 (RAF) を使用して、操作を実行します。このディレクティブの使用法および実行の流れは、obj.conf ファイルの NSAPI ディレクティブおよびサーバーアプリケーション関数 (SAF) とよく似ています。NSAPI、SAF と同様に、*pblocks* とも呼ばれるパラメータブロックを使用して、データは格納および送信されます。

ロボットディレクティブ、または RAF クラスには次の 6 つがあり、[320 ページの「フィルタリング処理」](#)に示されるフィルタリングのフェーズや操作に対応しています。

- セットアップ
- メタデータ
- データ
- 列挙
- 生成
- シャットダウン

各ディレクティブは、独自のロボットアプリケーション関数を持っています。たとえば、Metadata ディレクティブおよび Data ディレクティブではフィルタリング関数、Enumerate ディレクティブでは列挙関数、Generate ディレクティブでは生成関数を使用します。

ロボットアプリケーションの組み込み関数と、独自のロボットアプリケーション関数を記述する手順については、『Sun Java System Portal Server 6 2004Q2 Developer's Guide』を参照してください。

フィルタの記述または修正

ほとんどの場合、フィルタを一から記述する必要はありません。管理コンソールを使用すると、ほとんどのフィルタを作成できます。修正が必要な場合は、`filter.conf` ファイルおよび `filterrules.conf` ファイルを変更することができます。これらのファイルはディレクトリ `/var/opt/SUNWps/http-hostname-domain/portal` に存在します。

ただし、より複雑なパラメータのセットを作成する場合、ロボットが使用する設定ファイルを編集する必要があります。

フィルタの記述または変更の際は、次の点に注意してください。

- ディレクティブの実行順序 (特に各フェーズで使用可能な情報)
- ルールの順序

`robot.conf` ファイルで変更できるパラメータ、`filter.conf` ファイルで使用できるロボットアプリケーション関数、および独自のロボットアプリケーション関数の作成方法については、『Sun Java System Portal Server 6 2004Q2 Developer's Guide』を参照してください。

ユーザーが変更できるパラメータ

`robot.conf` ファイルは、`filter.conf` にある適切なフィルタにロボットを指定するなど、ロボット用の多くのオプションを定義します。旧バージョンとの下位互換性のために、`robot.conf` にシード URL を含めることもできます。

管理コンソールを使用すれば、ほとんどのパラメータを設定できるので、通常は `robot.conf` ファイルを編集する必要はありません。ただし、上級ユーザーはこのファイルを手動で編集して、管理コンソールからは設定できないパラメータを設定できます。このファイルの例については、[331 ページの「robot.conf ファイルの例」](#)を参照してください。

325 ページの表 14-3 は、robot.conf ファイルでユーザーが変更可能なパラメータを示しています。このテーブルの最初の列はパラメータ、2 番目の列はパラメータの説明、3 番目の列は例を示します。

表 14-3 ユーザーが変更できるパラメータ

パラメータ	説明	例
auto-proxy	ロボットのプロキシ設定を指定します。これは、プロキシサーバー、またはプロキシを自動的に設定する Java スクリプトファイルです。詳細については、『Sun Java System Portal Server 6 2005Q1 管理ガイド』を参照してください。	auto-proxy="http://proxy_server/proxy.pac"
bindir	ロボットがバインドディレクトリを PATH 環境に追加するかどうかを指定します。これは cmd-hook パラメータが指定するプログラムなど、ユーザーがロボットで外部プログラムを実行するための特別な PATH です。	bindir=path
cmd-hook	<p>ロボットが実行を完了するたびに実行される外部完了スクリプトを指定します。これは、コマンド名の完全パスで指定する必要があります。ロボットは /var/opt/SUNWps/ ディレクトリからこのスクリプトを実行します。</p> <p>デフォルトはありません。</p> <p>コマンドを実行するには、最低 1 つの RD が登録されている必要があります。</p> <p>完了スクリプトの記述については、『Sun Java System Portal Server 6 2004Q2 Developer's Guide』を参照してください。</p>	<p>cmd-hook="command-string"</p> <p>デフォルトはありません。</p>
command-port	<p>管理インタフェースまたはロボットコントロールパネルのような他のプログラムからコマンドを受け入れるために、ロボットが待機するソケットを指定します。</p> <p>セキュリティ上の理由から、remote-access が yes に設定されていない場合、ロボットはローカルホストからのコマンドだけを受け入れます。</p>	command-port=port_number

表 14-3 ユーザーが変更できるパラメータ (続き)

パラメータ	説明	例
connect-timeout	ネットワークが接続要求に応答する最大許容時間を指定します。 デフォルトは 120 秒です。	command-timeout=seconds
convert-timeout	ドキュメント変換に許可される最大時間を指定します。 デフォルトは 600 秒です。	convert-timeout=seconds
深さ	ロボットが調べるシード URL (開始ポイントとも呼ばれる) からのリンク数を指定します。このパラメータは、深さを指定しないシード URL のデフォルト値を設定します。 デフォルトは 10 です。 マイナス 1 の値 (depth=-1) は、リンクの深さが無限であることを表します。	depth=integer
email	ロボットを実行するユーザーの電子メールアドレスを指定します。 Web 管理者とサイトでロボットを実行するユーザーとが連絡できるように、電子メールアドレスは HTTP 要求ヘッダーで user-agent とともに送信されます。 デフォルトは user@domain です。	email=user@hostname
enable-ip	作成する各 RD の URL に IP アドレスを生成します。 デフォルトは true です。	enable-ip=[true yes false no]
enable-rdm-probe	サーバーが RDM をサポートするかどうかを確認し、ロボットはこのパラメータを使用して、検出する各サーバーに照会するかどうかを決定します。サーバーが RDM をサポートする場合は、サーバーは独自のリソース記述サーバーの役割を果たすことができるため、ロボットはこのサーバーのリソースを列挙しません。 デフォルトは false です。	enable-rdm-probe=[true false yes no]

表 14-3 ユーザーが変更できるパラメータ (続き)

パラメータ	説明	例
enable-robots-txt	<p>ロボットがアクセスする各サイトで、robots.txt ファイルが使用できる場合、このファイルをロボットが検査する必要があるかどうかを指定します。</p> <p>デフォルトは yes です。</p>	enable-robots-txt=[true false yes no]
engine-concurrent	<p>ロボットが使用する事前作成のスレッド数を指定します。</p> <p>デフォルトは 10 です。</p> <p>このパラメータは、管理コンソールから対話形式で設定することはできません。</p>	engine-concurrent=[1..100]
enumeration-filter	<p>リソースを列挙すべきかどうかを確認する場合に、ロボットが使用する列挙フィルタを指定します。値は filter.conf ファイルで定義されるフィルタ名にする必要があります。</p> <p>デフォルトは enumeration-default です。</p> <p>このパラメータは、管理コンソールから対話形式で設定することはできません。</p>	enumeration-filter=enumfilter name
generation-filter	<p>リソース用にリソース記述を生成する必要があるかどうかを確認する場合に、ロボットが使用する生成フィルタを指定します。値は filter.conf ファイルで定義されるフィルタ名にする必要があります。</p> <p>デフォルトは generation-default です。</p> <p>このパラメータは、管理コンソールから対話形式で設定することはできません。</p>	generation-filter=genfiltername
index-after-ngenerated	<p>検索エンジン用にバッチ処理する前に、ロボットが RD を収集する時間を分単位で指定します。</p> <p>このパラメータを指定しない場合は 256 分に設定されます。</p>	index-after-ngenerated=30

表 14-3 ユーザーが変更できるパラメータ (続き)

パラメータ	説明	例
loglevel	<p>ロギングのレベルを指定します。loglevel の値は次のように指定されます。</p> <ul style="list-style-type: none"> レベル 0: 重大なエラー以外は記録しません レベル 1: RD 生成も記録します (デフォルト) レベル 2: 検索状況も記録します レベル 3: フィルタリング状況も記録します レベル 4: 生成状況も記録します レベル 5: 検索の進行状況も記録します <p>デフォルト値は 1 です。</p>	loglevel=[0...100]
max-connections	<p>ロボットが実行できる並行検索の最大数を指定します。</p> <p>デフォルトは 8 です。</p>	max-connections=[1..100]
max-filesize-kb	<p>ロボットが検索するファイルの最大ファイルサイズを K バイト単位で指定します。</p>	max-filesize-kb=1024
max-memory-per-url / max-memory	<p>各 URL が使用する最大メモリをバイト単位で指定します。URL がより多くのメモリを必要とする場合、RD はディスクに保存されます。</p> <p>デフォルトは 1 です。</p> <p>このパラメータは、管理コンソールから対話形式で設定することはできません。</p>	max-memory-per-url=n_bytes
max-working	<p>ロボット稼働セットのサイズ、すなわちロボットが一度に処理できる URL の最大数を指定します。</p> <p>このパラメータは、管理コンソールから対話形式で設定することはできません。</p>	max-working=1024

表 14-3 ユーザーが変更できるパラメータ (続き)

パラメータ	説明	例
onCompletion	<p>実行完了後のロボットの動作を決定します。ロボットはアイドルモードになるか、ループバック、再起動、または終了することができます。</p> <p>デフォルトは <code>idle</code> です。</p> <p>このパラメータは <code>cmd-hook</code> パラメータとともに使用します。実行後のロボットは、<code>onCompletion</code> アクションの次に、<code>cmd-hook</code> プログラムを実行します。</p>	<code>OnCompletion=[idle loop quit]</code>
password	<p><code>httpd</code> 認証および <code>ftp</code> 接続に使用されるパスワードを指定します。</p>	<code>password=string</code>
referer	<p>Web ページにアクセスする際にロボットを <code>Referer</code> として識別するように設定されている場合、<code>HTTP</code> 要求に送信されるパラメータを指定します。</p>	<code>referer=string</code>
register-user and register-password	<p>検索エンジンデータベースに <code>RD</code> を登録するために使用されるユーザー名を指定します。</p> <p>このパラメータは、検索エンジン管理インタフェースから対話形式で設定することはできません。</p>	<code>register-user=string</code>
register-password	<p>検索エンジンデータベースに <code>RD</code> を登録するために使用されるパスワードを指定します。</p> <p>このパラメータは、管理コンソールから対話形式で設定することはできません。</p>	<code>register-password=string</code>
remote-access	<p>ロボットが、リモートホストからのコマンドを受け入れることができるかどうかを指定します。</p> <p>デフォルトは <code>false</code> です。</p>	<code>remote-access=[true false yes no]</code>
robot-state-dir	<p>ロボットが状態を保存するディレクトリを指定します。この作業ディレクトリで、ロボットは収集された <code>RD</code> の数などを記録できます。</p>	<code>robot-state-dir="/var/opt/SUNWps/instance/portal/robot"</code>

表 14-3 ユーザーが変更できるパラメータ (続き)

パラメータ	説明	例
server-delay	同じ Web サイトに次にアクセスするまでの時間間隔を指定します。この時間を指定することで、ロボットが頻繁に同じサイトにアクセスするのを防止できます。	server-delay=delay_in_seconds
site-max-connections	ロボットが任意の 1 つのサイトに対して行う同時接続の最大数を示します。 デフォルトは 2 です。	site-max-connections=[1..100]
smart-host-heuristics	ロボットが正規の DNS ホスト名を巡回するサイトを変更できます。たとえば、www123.siroe.com が www.siroe.com に変更されます。 デフォルトは false です。	smart-host-heuristics=[true false]
tmpdir	ロボットが一時ファイルを作成する場所を指定します。 環境変数 TMPDIR を設定する際に、この変数を使用します。	tmpdir=path
user-agent	http-request 内の電子メールアドレスとともにサーバーに送信されるパラメータを指定します。	user-agent=iPlanetRobot/4.0
username	ロボットを実行するユーザーのユーザー名を指定します。これは、http 認証および ftp 接続で使用されます。 デフォルトは anonymous です。	username=string

robot.conf ファイルの例

ここでは、robot.conf ファイルの例について説明します。この例でコメントが付いたパラメータには、表示されているデフォルト値が適用されます。最初のパラメータである csid は、このファイルを使用する検索エンジンインスタンスを示しています。このパラメータの値は変更しないでください。このファイルのパラメータの定義については、[324 ページの「ユーザーが変更できるパラメータ」](#)を参照してください。

注 このサンプルファイルには、検索エンジンによって使用されるいくつかのパラメータが含まれています。その中には csid パラメータのように変更してはならないものもあります。

```
<Process csid="x-catalog://budgie.siroe.com:80/jack" ¥
  auto-proxy="http://sesta.varrius.com:80/"
  auto_serv="http://sesta.varrius.com:80/"
  command-port=21445
  convert-timeout=600
  depth="-1"
  # email="user@domain"
  enable-ip=true
  enumeration-filter="enumeration-default"
  generation-filter="generation-default"
  index-after-ngenerated=30
  loglevel=2
  max-concurrent=8
  site-max-concurrent=2
  onCompletion=idle
  password=boots
  proxy-loc=server
  proxy-type=auto
  robot-state-dir="/var/opt/SUNWps/https-budgie.siroe.com/ ¥
  ps/robot"
  server-delay=1
  smart-host-heuristics=true
  tmpdir="/var/opt/SUNWps/https-budgie.siroe.com/ps/tmp"
  user-agent="iPlanetRobot/4.0"
  username=jack
</Process>
```

robot.conf ファイルの例

定義済みのロボットアプリケーション関数

この章では、Sun Java™ System Portal Server の検索エンジンで定義済みのロボットアプリケーション関数 (RAF) について説明し、パラメータ仕様および例を紹介します。これらの関数を `filter.conf` ファイルで使用し、フィルタ定義を作成および変更できます。ファイル `filter.conf` は、ディレクトリ `/var/opt/SUNWps/http-hostname-domain/portal/config` に配置されています。

ファイル `filter.conf` には、列挙フィルタおよび生成フィルタの定義が含まれます。これらの各フィルタは、`filterrules.conf` ファイルに格納されているルールセットを呼び出します。フィルタルールには両方のフィルタで使用するルールが含まれますが、フィルタ定義にはフィルタ固有の命令が含まれます。

フィルタルールの定義方法を理解するためには、ファイル `filterrules.conf` を調べます。通常は、管理コンソールを使用してフィルタルールを作成するため、このファイルを直接編集する必要はありません。

フィルタ定義の例を見るには、ファイル `filter.conf` を調べる必要があります。フィルタのためにリソースを生成しないでリソースを列挙するようロボットに指示するなど、管理コンソールに適応しない方法でフィルタを変更する場合、編集する必要があるのは `filter.conf` ファイルだけです。

この章で説明する内容は次のとおりです。

- [ソースとデスティネーション](#)
- [セットアップ関数](#)
- [フィルタリング関数](#)
- [サポート関数のフィルタリング](#)
- [列挙関数](#)
- [生成関数](#)
- [シャットダウン関数](#)

ソースとデスティネーション

ほとんどのロボットアプリケーション関数 (RAF) は情報のソースを必要とし、デスティネーションに送信するデータを生成します。ソースはロボット内で定義され、最終的に生成されるリソース記述のフィールドに関連している必要はありません。一方、デスティネーションは通常、リソース記述サーバーのスキーマによって定義されるリソース記述のフィールド名になります。

データベーススキーマを指定するために管理コンソールを使用する詳細については、[第 13 章「検索エンジンサービスの管理」](#)を参照してください。

次の節では、フィルタリングプロセスの各段階と、それらの段階で使用できるソースについて説明します。

セットアップ段階で使用可能なソース

セットアップ段階では、フィルタが設定されますが、リソースの URL またはコンテンツについての情報は取得できません。

メタデータフィルタリング段階で使用可能なソース

メタデータ段階ではロボットはリソースの URL を取得しますが、リソースのコンテンツはダウンロードされていないため `filter.conf` ファイルなどのほかのソースから導出されるデータと、URL についての情報が使用可能です。ただし、この段階では、リソースのコンテンツについての情報は使用できません。

[表 15-1](#) は、メタデータ段階で RAF で使用できるソースを示しています。この表では 3 つの列があります。最初の列はソース、2 番目の列は説明、3 番目の列は例を示します。

表 15-1 メタデータ段階で RAF で使用できるソース

ソース	説明	例
csid	カタログサーバー ID	x-catalog//budgie.siroe.com:8086/alexandria
depth	開始点からトラバースされたリンク数	10
enumeration filter	列挙フィルタの名前	enumeration1
generation filter	生成フィルタの名前	generation1
host	URL のホスト部分	home.siroe.com

表 15-1 メタデータ段階で RAF で使用できるソース (続き)

ソース	説明	例
IP	ホストの数値バージョン	198.95.249.6
protocol	URL のアクセス部分	http、https、ftp、file
path	URL のパス部分	/、/index.html、/documents/listing.html
URL	完全な URL	http://developer.siroe.com/docs/manuals/

データ段階で使用可能なソース

データ段階では、ロボットがその URL のリソースのコンテンツをダウンロードしているので、記述、作成者などのコンテンツに関するデータにアクセスできます。

リソースが HTML ファイルである場合、ロボットは HTML ヘッダーの <META> タグをパースします。したがって、<META> タグに含まれるデータは、データ段階で使用可能です。

データ段階では、メタデータ段階で使用可能なソースのほかに、次のソースを RAF で使用できます。この表では 3 つの列があります。最初の列はソース、2 番目の列は説明、3 番目の列は例を示します。

表 15-2 データ段階で RAF で使用できるソース

ソース	説明	例
content-charset	リソースが使用する文字セット	
content-encoding	エンコードの形式	
content-length	リソースのバイト単位のサイズ	
content-type	リソースの MIME タイプ	text/html、image/jpeg
expires	リソースが期限切れになる日付	
last-modified	リソースの最終更新日	
<META> タグ内のデータ	HTML リソースのヘッダーの <META> タグで提供されるデータ	作成者 記述 キーワード

これらのすべてのソース (<META> タグのデータを除く) は、リソースの取得時に返される HTTP 応答ヘッダーから導出されます。

列挙、生成、およびシャットダウン段階で使用可能なソース

列挙および生成の段階では、データ段階と同じデータソースが使用可能です。

シャットダウン段階では、フィルタはフィルタリングを完了し、シャットダウンします。この段階用に書き込まれた関数は、データ段階で使用可能なソースと同じデータソースを使用できますが、シャットダウン関数の動作は通常、状況の終了およびクリーンアップに制限されます。

有効化パラメータ

各関数には有効化パラメータがあります。値には、true、false、on、または off を指定できます。管理コンソールは、これらのパラメータを使用して特定のディレクトイブを有効または無効にします。

次の例は text/html の列挙を有効にし、text/plain の列挙を無効にします。

```
# Perform the enumeration on HTML only
Enumerate enable=true fn=enumerate-urls max=1024 type=text/html
Enumerate enable=false fn=enumerate-urls-from-text max=1024
type=text/plain
```

enable=false パラメータまたは enable=off パラメータの追加は、行をコメントアウトするのと同じ効果があります。管理インタフェースは、コメントを書き込みませんが、代わりに enable パラメータを書き込みます。

セットアップ関数

ここでは、列挙および生成フィルタの両方によってセットアップ段階中に使用される関数について説明します。次の関数について説明します。

- [337 ページの「`filterrules-setup`」](#)
- [337 ページの「`setup-regex-cache`」](#)
- [338 ページの「`setup-type-by-extension`」](#)

filterrules-setup

`filterrules-setup` 関数を使用する場合、`logtype` は使用するログファイルのタイプになります。値には `verbose`、`normal`、または `terse` を指定できます。

パラメータ

表 15-3 は、`filterrules-setup` 関数で使用されるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-3 `filterrules-setup` パラメータ

パラメータ	説明
<code>config</code>	このフィルタが使用するフィルタルールを含むファイルのパス名。

例

```
Setup fn=filterrules-setup config=./config/filterrules.conf
logtype=normal
```

setup-regex-cache

`setup-regex-cache` 関数は、`filter-by-regex` および `generate-by-regex` 関数のキャッシュサイズを初期化します。デフォルト値である 32 以外の数値を指定するときは、この関数を使用します。

パラメータ

表 15-4 は、`setup-regex-cache` 関数で使用されるパラメータを示しています。この表では 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-4 `setup-regex-cache` パラメータ

パラメータ	説明
<code>cache-size</code>	<code>regex</code> キャッシュに保持されるコンパイル済み正規表現の最大数。

例

```
Setup fn=setup-regex-cache cache-size=28
```

setup-type-by-extension

`setup-type-by-extension` 関数はファイル名の拡張子を認識するようにフィルタを設定します。この関数を呼び出してから、`assign-type-by-extension` 関数を使用します。パラメータとして指定されたファイルは、標準の MIME コンテンツタイプとファイル拡張子の文字列の間のマッピングを含む必要があります。

パラメータ

表 15-5 は、`setup-type-by-extension` 関数で使用されるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-5 `setup-type-by-extension` パラメータ

パラメータ	説明
<code>file</code>	MIME タイプ設定ファイルの名前。

例

```
Setup fn=setup-type-by-extension file=./config/mime.types
```

フィルタリング関数

次の関数はメタデータおよびデータの段階で動作し、関数およびそのパラメータによって指定された特定の条件に基づいて、リソースを許可または拒否します。

これらの関数は、`filter.conf` ファイルの列挙フィルタおよび生成フィルタの両方で使用できます。

これらの各「`filter-by`」関数は比較を行い、リソースを許可または拒否します。リソースが許可されると、処理が次のフィルタリング段階へ進みます。リソースが拒否されると、リソースが列挙または生成を継続する条件を満たしていないという理由で処理が停止します。次の関数について説明します。

- [filter-by-exact](#)
- [filter-by-max](#)
- [filter-by-md5](#)
- [filter-by-prefix](#)
- [filter-by-regex](#)
- [filterrules-process](#)

filter-by-exact

`filter-by-exact` 関数は、`allow/deny` 文字列が情報のソースに完全に一致する場合、リソースを許可または拒否します。キーワード `all` は任意の文字列と一致します。

パラメータ

表 15-6 は、`filter-by-exact` 関数で使われるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-6 `filter-by-exact` パラメータ

パラメータ	説明
<code>src</code>	情報のソース。
<code>allow/deny</code>	文字列を含みます。

例

次の例は、`content-type` が `text/plain` のリソースすべてをフィルタして取り除きます。これにより、ほかのすべてのリソースの処理が続行されます。

```
Data fn=filter-by-exact src=type deny=text/plain
```

filter-by-max

filter-by-max 関数では、指定された情報ソースが指定値以下の場合にリソースが許可されます。情報ソースが指定された値よりも大きい場合は、リソースは拒否されません。

この関数は、1つのフィルタについて1回しか呼び出すことができません。

パラメータ

表 15-7 は、filter-by-max 関数で使用されるパラメータを示しています。この表には2つの列があります。最初の列はパラメータ、2番目の列は説明を示します。

表 15-7 filter-by-max パラメータ

パラメータ	説明
src	情報のソース。hosts、objects、または depth のどれかの値を指定する必要があります。
value	比較のための値を指定します。

例

この例は、content-length が 1024K バイトより小さいリソースを許可します。

```
MetaData fn-filter-by-max src=content-length value=1024
```

filter-by-md5

filter-by-md5 関数は、指定の MD5 チェックサム値を持つ最初のリソースだけを許可します。このロボットによって、現在のリソースの MD5 が以前のリソースで検出された場合、現在のリソースは拒否されます。これにより、複数の URL を持つ同一のリソースまたは単一のリソースの重複を防ぎます。

この関数は、データ段階またはそれ以降の段階でだけ呼び出すことができます。この関数は、1つのフィルタについて1回しか呼び出すことができません。フィルタは generate-md5 関数を呼び出し、filter-by-md5 を呼び出す前に MD5 チェックサムを生成する必要があります。

パラメータ

none

例

次の例は、MD5 チェックサムを処理する一般的な方法を示します。最初にチェックサムを生成し、そのチェックサムに基づいてフィルタリングします。

```
Data fn=generate-md5
```

```
Data fn=filter-by-md5
```

filter-by-prefix

filter-by-prefix 関数は、指定の情報ソースが指定されたプレフィックス文字列で開始する場合、リソースを許可または拒否します。リソースは、完全に一致する必要はありません。キーワード all は任意の文字列と一致します。

パラメータ

表 15-8 は、filter-by-prefix 関数で 사용되는パラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-8 filter-by-prefix パラメータ

パラメータ	説明
src	情報のソース。
allow/deny	プレフィックス比較の文字列を含みます。

例

次の例は、content-type が text/html や text/plain など、任意の種類テキストであるリソースを許可します。

```
MetaData fn=filter-by-prefix src=type allow=text
```

filter-by-regex

filter-by-regex 関数は、正規表現によるパターンマッチングをサポートします。この関数は、指定の正規表現に一致するリソースを許可します。使用可能な正規表現構文は、POSIX.1 仕様によって定義されます。¥¥* の正規表現は、任意の文字に一致します。

パラメータ

表 15-9 は、filter-by-regex 関数で 사용되는パラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-9 filter-by-regex パラメータ

パラメータ	説明
src	情報のソース。

表 15-9 filter-by-regex パラメータ (続き)

パラメータ	説明
allow/deny	正規表現の文字列を含みます。

例

次の例は、政府 (gov) ドメインのサイトからのすべてのリソースを拒否します。

```
MetaData fn=filter-by-regex src=host deny=¥¥*.gov
```

filterrules-process

filterrules-process 関数は、filterrules.conf ファイルのルールで処理します。

パラメータ

none

例

```
MetaData fn=filterrules-process
```

サポート関数のフィルタリング

次の関数は、リソース上で情報を操作または生成するため、フィルタリング中に使用されます。ロボットは、フィルタリング関数を呼び出すことによって、リソースを処理することができます。これらの関数は、ファイル filter.conf の列挙および生成のフィルタで使用することができます。次の関数について説明します。

- [assign-source](#)
- [assign-type-by-extension](#)
- [clear-source](#)
- [convert-to-html](#)
- [copy-attribute](#)
- [generate-by-exact](#)
- [generate-by-prefix](#)
- [generate-by-regex](#)
- [generate-md5](#)

- [generate-rd-expires](#)
- [generate-rd-last-modified](#)
- [rename-attribute](#)

assign-source

`assign-source` 関数は、指定した情報ソースに新規の値を割り当てます。これにより、フィルタリング処理中の編集が可能になります。この関数は、明示的な新規値を割り当てるか、または、別の情報ソースから値をコピーすることができます。

パラメータ

表 15-10 は、`assign-source` 関数で使用するパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-10 `assign-source` パラメータ

パラメータ	説明
<code>dst</code>	値を変更するソースの名前。
<code>value</code>	明示的な値を指定。
<code>src</code>	<code>dst</code> にコピーする情報ソース。

`value` パラメータまたは `src` パラメータのどちらかを指定する必要がありますが、両方は指定しません。

例

```
Data fn=assign-source dst=type src=content-type
```

assign-type-by-extension

`assign-type-by-extension` 関数は、リソースのファイル名を使用してタイプを決定し、処理を継続するリソースにこのタイプを割り当てます。

`assign-type-by-extension` を使用可能にするには、セットアップ中に `setup-type-by-extension` 関数を呼び出す必要があります。

パラメータ

表 15-11 は、`assign-type-by-extension` 関数で使われるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-11 `assign-type-by-extension` パラメータ

パラメータ	説明
<code>src</code>	比較するファイル名のソース。ソースを指定しない場合、デフォルトではリソースのパスになります。

例

```
MetaData fn=assign-type-by-extension
```

clear-source

`clear-source` 関数は、指定されたデータソースを削除します。通常は、この関数を実行する必要はありません。`assign-source` を使用することにより、ソースを作成および置換することができます。

パラメータ

表 15-12 は、`clear-source` 関数で使われるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-12 `clear-source` パラメータ

パラメータ	説明
<code>src</code>	削除するソースの名前。

例

次の例は、パスのソースを削除します。

```
MetaData fn=clear-source src=path
```

convert-to-html

`convert-to-html` 関数は、リソースのタイプが指定の MIME タイプと一致する場合、現在のリソースを HTML ファイルに変換して処理を続けます。変換フィルタは、変換中のファイルのタイプを自動的に検出します。

パラメータ

表 15-13 は、`convert-to-html` 関数で使用されるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-13 `convert-to-html` パラメータ

パラメータ	説明
<code>type</code>	変換を行う MIME タイプ。

例

次の関数呼び出しのシーケンスにより、フィルタは、すべての Adobe Acrobat PDF ファイル、Microsoft RTF ファイル、および FrameMaker MIF ファイル、およびファイルを配信するサーバーによってタイプが指定されていないファイルを HTML に変換します。

```
Data fn=convert-to-html type=application/pdf
```

```
Data fn=convert-to-html type=application/rtf
```

```
Data fn=convert-to-html type=application/x-mif
```

```
Data fn=convert-to-html type=unknown
```

copy-attribute

`copy-attribute` 関数は、リソース記述の 1 つのフィールドからの値を別のフィールドにコピーします。

パラメータ

表 15-14 は、`copy-attribute` 関数で使用されるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-14 `copy-attribute` パラメータ

パラメータ	説明
<code>src</code>	コピー元のリソース記述のフィールド。
<code>dst</code>	ソースのコピー先のリソース記述の項目。

表 15-14 copy-attribute パラメータ (続き)

パラメータ	説明
truncate	コピーするソースの最大長。
clean	省略されたテキストを修正するかどうか (分割された単語を修正するかどうかなど) を示す論理値パラメータ。デフォルトでは、このパラメータは false に設定されています。

例

```
Generate fn=copy-attribute ¥
src=partial-text dst=description truncate=200 clean=true
```

generate-by-exact

generate-by-exact 関数は、既存のソースが別の値と完全に一致する場合に限り、指定された値を持つソースを生成します。

パラメータ

表 15-15 は、generate-by-exact 関数で使用されるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-15 generate-by-exact パラメータ

パラメータ	説明
dst	生成するソースの名前。
value	dst に割り当てる値。
src	比較対象のソース。

例

次の例は、ホストが www.siroe.com の場合に、分類を Siroe に設定します。

```
Generate fn="generate-by-exact" match="www.siroe.com:80" src="host"
value="Siroe" dst="classification"
```

generate-by-prefix

この `generate-by-prefix` 関数は、既存のソースのプレフィックスが別の値と一致する場合に限り、指定された値を持つソースを生成します。

パラメータ

表 15-16 は、`generate-by-prefix` 関数で使用されるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-16 `generate-by-prefix` パラメータ

パラメータ	説明
<code>dst</code>	生成するソースの名前。
<code>value</code>	<code>dst</code> に割り当てる値。
<code>src</code>	比較対象のソース。
<code>match</code>	<code>src</code> と比較する値。

例

次の例は、プロトコルプレフィックスが `HTTP` の場合、分類を `World Wide Web` に設定します。

```
Generate fn="generate-by-prefix" match="http" src="protocol"
value="World Wide Web" dst="classification"
```

generate-by-regex

`generate-by-regex` 関数は、既存のソースが正規表現に一致する場合に限り、指定した値を持つソースを生成します。

パラメータ

表 15-17 は、`generate-by-regex` 関数で使用されるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-17 `generate-by-regex` パラメータ

パラメータ	説明
<code>dst</code>	生成するソースの名前。
<code>value</code>	<code>dst</code> に割り当てる値。
<code>src</code>	比較対象のソース。
<code>match</code>	<code>src</code> と比較する正規表現の文字列。

例

次の例は、ホスト名が正規表現 `*.siroe.com` に一致する場合に、分類を `Siroe` に設定します。たとえば、`developer.siroe.com` および `home.siroe.com` の両方にあるリソースは、`Siroe` として分類されます。

```
Generate fn="generate-by-regex" match="¥¥*.siroe.com" src="host"
value="Siroe" dst="classification"
```

generate-md5

`generate-md5` 関数は、MD5 チェックサムを生成し、リソースに追加します。これにより、`filter-by-md5` 関数を使用して、重複する MD5 チェックサムを持つリソースを拒否することができます。

パラメータ

none

例

```
Data fn=generate-md5
```

generate-rd-expires

`generate-rd-expires` 関数は、有効期限を生成して、指定されたソースに追加します。この関数は、HTTP ヘッダーおよび HTML `<META>` タグなどのメタデータを使用し、リソースから任意の有効期限を取得します。有効期限が存在しない場合は、現在の日付から 3 ヶ月間の有効期限を生成します。

パラメータ

表 15-18 は、`generate-rd-expires` 関数で 사용되는パラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-18 `generate-rd-expires` パラメータ

パラメータ	説明
<code>dst</code>	ソースの名前。指定を省略した場合、デフォルトは <code>rd-expires</code> となります。

例

```
Generate fn=generate-rd-expires
```

generate-rd-last-modified

generate-rd-last-modified 関数は、現在の時間を指定されたソースに追加します。

パラメータ

表 15-19 は、generate-rd-last-modified 関数で使用されるパラメータを示しています。この表には2つの列があります。最初の列はパラメータ、2番目の列は説明を示します。

表 15-19 generate-rd-last-modified パラメータ

パラメータ	説明
dst	ソースの名前。指定を省略した場合、デフォルトは rd-last-modified となります。

例

```
Generate fn=generate-last-modified
```

rename-attribute

rename-attribute 関数は、リソース記述のフィールドの名前を変更します。たとえば、[extract-html-meta](#) で <META> タグからフィールドに情報をコピーし、そのフィールドの名前を変更する場合にとっても便利です。

パラメータ

表 15-20 は、rename-attribute 関数で使用されるパラメータを示しています。この表には2つの列があります。最初の列はパラメータ、2番目の列は説明を示します。

表 15-20 rename-attribute パラメータ

パラメータ	説明
src	1つの名前から別の名前へのマッピングを含む文字列。

例

次の例は、属性の名前を author から author-name に変更します。

```
Generate fn=rename-attribute src="author->author-name"
```

列挙関数

次の関数は、列挙段階で動作します。これらの関数は、ロボットが指定のリソースからリンクを収集するかどうか、およびその方法を制御し、リソースの発見を継続する開始点として使用します。ここでは、次の関数について説明します。

- [enumerate-urls](#)
- [enumerate-urls-from-text](#)

enumerate-urls

`enumerate-urls` 関数はリソースをスキャンし、ハイパーテキストリンクで検出されるすべての URL を列挙します。結果を使用して、リソース発見を継続します。コンテンツタイプを指定し、列挙される URL の種類を制限することができます。

パラメータ

表 15-21 は、`enumerate-urls` 関数で 사용되는パラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-21 `enumerate-urls` パラメータ

パラメータ	説明
<code>max</code>	指定のリソースから生成する URL の最大数。指定を省略した場合、デフォルトは 1024 となります。
<code>type</code>	指定したコンテンツタイプの URL に列挙を制限するコンテンツタイプ。 <code>type</code> は、オプションのパラメータです。指定を省略した場合、すべての URL が列挙されます。

例

次の例は、HTML URL だけを、最大値 1024 まで列挙します。

```
Enumerate fn=enumerate-urls type=text/html
```

enumerate-urls-from-text

`enumerate-urls-from-text` 関数はテキストリソースをスキャンし、正規表現 `URL:.*` に一致する文字列を検索します。これはロボットを生成し、これらの文字列から URL を列挙し、さらにリソース記述を生成します。

パラメータ

表 15-22 は、`enumerate-urls-from-text` 関数で 사용되는パラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-22 `enumerate-urls-from-text` パラメータ

パラメータ	説明
<code>max</code>	指定のリソースから生成する URL の最大数。指定を省略した場合、デフォルトは 1024 となります。

例

```
Enumerate fn=enumerate-urls-from-text
```

生成関数

次の関数は、フィルタリングの生成段階で使用されます。生成関数は、リソース記述に入る情報を生成することができます。通常、リソースの本体から情報を抽出するか、リソースのメタデータから情報をコピーします。ここでは、次の関数について説明します。

- [extract-full-text](#)
- [extract-html-meta](#)
- [extract-html-text](#)
- [extract-html-toc](#)
- [extract-source](#)
- [harvest-summarizer](#)

extract-full-text

`extract-full-text` 関数は、リソースの完全なテキストを抽出して、リソース記述に追加します。

注 `extract-full-text` 関数の使用には注意が必要です。リソース記述のサイズを非常に大きくすることができるため、データベースが膨大化したり、ネットワーク帯域幅に全体的な悪影響を及ぼしたりする可能性があります。

パラメータ

表 15-23 は、`extract-full-text` 関数で使用するパラメータを示します。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-23 `extract-full-text` パラメータ

パラメータ	説明
<code>truncate</code>	リソースから抽出する文字の最大数。
<code>dst</code>	完全なテキストを受信するスキーマ項目の名前。

例

```
Generate fn=extract-full-text
```

extract-html-meta

`extract-html-meta` 関数は、HTML ファイルから任意の `<META>` または `<TITLE>` 情報を抽出し、リソース記述に追加します。コンテンツタイプの指定により、生成する URL の種類が制限される場合があります。

パラメータ

表 15-24 は、`extract-html-meta` 関数で使用するパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-24 `extract-html-meta` パラメータ

パラメータ	説明
<code>truncate</code>	抽出する最大バイト数。
<code>type</code>	省略可能なパラメータ。指定を省略した場合は、すべての URL が生成されます。

例

```
Generate fn=extract-html-meta truncate=255 type=text/html
```

extract-html-text

`extract-html-text` 関数は、HTML タグを除いた、テキストの最初の数文字を HTML ファイルから抽出し、そのテキストをリソース記述に追加します。これにより、ドキュメントのテキストの最初の部分を RD に含むことができます。コンテンツタイプの指定により、生成する URL の種類が制限される場合があります。

パラメータ

表 15-25 は、`extract-html-text` 関数で使われるパラメータを示します。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-25 `extract-html-text` パラメータ

パラメータ	説明
<code>truncate</code>	抽出する最大バイト数。
<code>skip-headings</code>	<code>true</code> に設定すると、ドキュメント内で発生する任意の HTML ヘッダーは無視されます。
<code>type</code>	省略可能なパラメータ。指定を省略した場合は、すべての URL が生成されます。

例

```
Generate fn=extract-html-text truncate=255 type=text/html
skip-headings=true
```

extract-html-toc

`extract-html-toc` 関数は、HTML ヘッダーから `table-of-contents` を抽出し、リソース記述に追加します。

パラメータ

表 15-26 は、`extract-html-toc` 関数で使われるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-26 `extract-html-toc` パラメータ

パラメータ	説明
<code>truncate</code>	抽出する最大バイト数。

表 15-26 extract-html-toc パラメータ (続き)

パラメータ	説明
level	抽出する HTML ヘッダーの最大レベル。このパラメータは、目次の深さを制御します。

例

```
Generate fn=extract-html-toc truncate=255 level=3
```

extract-source

extract-source 関数は、指定のソースから特定の値を抽出し、リソース記述に追加します。

パラメータ

表 15-27 は、extract-source 関数で使われるパラメータを示しています。この表には 2 つの列があります。最初の列はパラメータ、2 番目の列は説明を示します。

表 15-27 extract-source パラメータ

パラメータ	説明
src	ソース名のリスト。-> 演算子を使用し、RD 属性の新規名を定義できます。たとえば、type->content-type は type という名前のソースの値を取得し、content-type という名前の属性で RD に保存されます。

例

```
Generate fn=extract-source
src="md5,depth,rd-expires,rd-last-modified"
```

harvest-summarizer

harvest-summarizer 関数は、リソース上で Harvest サマライザを実行し、結果をリソース記述に追加します。

Harvest サマライザを実行するには、ロボットを実行する前に、path に \$HARVEST_HOME/lib/gatherer を設定する必要があります。

パラメータ

表 15-28 は、harvest-summarizer 関数で使用されるパラメータを示しています。この表には2つの列があります。最初の列はパラメータ、2番目の列は説明を示します。

表 15-28 harvest-summarizer パラメータ

パラメータ	説明
summarizer	サマライザプログラムの名前。

例

```
Generate fn-harvest-summarizer summarizer=HTML.sum
```

シャットダウン関数

次の関数は、列挙および生成の両関数によってシャットダウン段階中に使用できます。

filterrules-shutdown

ルールが実行されたあと、filterrules-shutdown 関数はクリーンアップおよびシャットダウンを実行します。

パラメータ

none

例

```
Shutdown fn=filterrules-shutdown
```

シャットダウン関数

登録サービスの管理

この章で説明する内容は次のとおりです。

- [概要](#)
- [登録サービスの管理](#)
- [登録チャンネルの使用](#)

概要

登録サービスを利用することで、ユーザーは多くの情報を網羅する興味対象のプロファイルを作成できます。このリリースでサポートされる情報ソースには、カテゴリ、ディスカッション、検索可能ドキュメントが含まれます。ユーザーが登録チャンネルにアクセスするたびに、プロファイルは最新の情報で更新されます。登録チャンネルは、ユーザーが定義したカテゴリ別ドキュメントやディスカッションのプロファイルエントリと一致するヒット回数 (関連情報) を要約します。

検索サービスは、次のように使用されます。

- 指定日数の範囲内にターゲットカテゴリで一致する新規ドキュメントの検索とカウント
- 指定日数の範囲内にディスカッションで一致する新規関連コメントの検索とカウント
- 保存されている検索に対するドキュメントの検索とカウント

結果は、一致情報の数を示すリンクとしてプロファイルエントリに表示されます。ユーザーがこのリンクをクリックすると、その一致自体の詳細が表示されます。

カテゴリ登録では、ユーザーがリンクをクリックすると検索チャンネルが表示されます。ここには、興味対象の特定のドキュメントが標準のカテゴリ検索結果の形式で要約されます。登録チャンネルは、より詳細な情報への入り口として機能します。

プロファイラ機能により、指定した興味対象のコンテンツが変更された場合に、メールで通知できます。プロファイラは、Access Manager からユーザーの登録情報を入手し、検索サービスから結果を取り込んで、電子メール通知をユーザーに送信します。プロファイラは、スケジュール設定して組織レベルで特定の時間に実行できます。

登録サービスの管理

管理者は、登録サービスを有効化、または無効化することができます。登録サービスは、次のレベルで管理できます。

- ルートレベル
- 組織レベル
- 組織ユーザーレベル

ルートレベル

登録サービスのルートレベルの管理では、システム全体を対象に、タイプ別 (カテゴリ、ディスカッション、保存される検索) の登録のデフォルト最大数を設定します。ルートレベルでの登録サービスの定義については、[359 ページの「ルートレベルで登録サービスを定義する」](#)を参照してください。

組織レベル

登録サービスの組織レベルの管理では、システム全体を対象に、タイプ別 (カテゴリ、ディスカッション、保存される検索) の登録のデフォルト最大数を上書きします。組織レベルでの登録サービスの定義については、[359 ページの「組織レベルで登録サービスを定義する」](#)を参照してください。

組織ユーザーレベル

組織ユーザーレベルでの登録サービスの管理では、ユーザーの登録サービスの設定を編集します。管理者は、ユーザーのサービスデータを次のように管理できます。

- ユーザーの登録の更新
- ユーザーの登録の削除

ユーザーの登録サービスの管理については、[360 ページの「ユーザーの登録サービスを管理する」](#)を参照してください。

ルートレベルで登録サービスを定義する

1. Sun Java System Access Manager 管理コンソールにログインし、「サービス設定」タブを選択します。
2. 左のパネルのリストから「登録」サービスを選択します。
3. 次の項目のデフォルト値を変更します。
 - **カテゴリ登録の最大数**：ユーザーが登録できるカテゴリの最大数を指定します。
 - **ディスカッション登録の最大数**：ユーザーが登録できるディスカッションの最大数を指定します。
 - **保存される検索の最大数**：保存できる検索の最大数を指定します。
4. 次のボタンを選択します。
 - 指定した値を保存するときは「保存」。
 - 変更した値を元に戻すときは「リセット」。

組織レベルで登録サービスを定義する

1. Sun Java System Access Manager 管理コンソールにログインし、組織の「表示」プルダウンメニューから「サービス」を選択します。
2. 左のパネルのリストから「登録」サービスを選択します。
3. プロファイラを開始する曜日と時刻（時間と分）を選択します。
4. プロファイラを停止する曜日と時刻（時間と分）を選択します。
5. 次の項目のデフォルト値を変更します。
 - **競合の解決レベル**：「最も高い」、「高め」、「高」、「中」、「低」、「低め」、「最低」に設定できます。
 - **カテゴリ登録の最大数**：ユーザーが登録できるカテゴリの最大数を指定します。
 - **ディスカッション登録の最大数**：ユーザーが登録できるディスカッションの最大数を指定します。
 - **保存される検索の最大数**：保存できる検索の最大数を指定します。
6. 次のボタンを選択します。
 - 指定した値を保存するときは「保存」。
 - 変更した値を元に戻すときは「リセット」。
 - 「削除」。

ユーザーの登録サービスを管理する

1. 管理コンソールにログインし、組織の「表示」プルダウンメニューから「ユーザー」を選択します。
2. 「ユーザー」を選択します。
右のパネルにユーザー情報が表示されます。
3. 「表示」プルダウンメニューから「登録」を選択します。
ユーザーの登録を編集するためのページが表示されます。
4. 登録の定義を編集します。
登録のタイプごとに、登録を追加または消去します。形式は次のとおりです。

○ カテゴリ登録

label | target category | scope | lapsed time

各表記の意味は次のとおりです。

label	編集される登録に与えられる論理参照で、文字列として指定する必要があります。必須フィールド。
target category	ABC:DEF:GHI という形式の文字列で指定する必要があります。
scope	検索クエリー。検索演算子を含む有効な検索文字列として指定する必要があります。
lapsed time	次のいずれかの値を指定する必要があります。 <ul style="list-style-type: none"> • 0 = すべて • 7 = 1 週間以内 • 30 = 1 ヶ月以内 • 180 = 6 ヶ月以内 • 365 = 1 年以内

○ ディスカッション登録

label | target discussion RD's URL | scope | lapsed time | minimum rating

各表記の意味は次のとおりです。

label	編集される登録に与えられる論理参照で、文字列として指定する必要があります。必須フィールド。
-------	---

target discussion RD's URL	ディスカッションの URL と一致する文字列を指定する必要があります。ディスカッションの編集時に、ユーザーは登録チャンネルを使用してこの値を変更することができません。
scope	検索クエリー。検索演算子を含む有効な検索文字列として指定する必要があります。
lapsed time	次のいずれかの値を指定する必要があります。 <ul style="list-style-type: none"> • 0 = すべて • 7 = 1 週間以内 • 30 = 1 ヶ月以内 • 180 = 6 ヶ月以内 • 365 = 1 年以内
minimum rating	最小重要度に基づくフィルタです。

○ 保存される検索

label | target category | scope | lapsed time

各表記の意味は次のとおりです。

label	編集される登録に与えられる論理参照で、文字列として指定する必要があります。必須フィールド。
target category	ABC:DEF:GHI という形式の文字列で指定する必要があります。
scope	検索クエリー。検索演算子を含む有効な検索文字列として指定する必要があります。
lapsed time	次のいずれかの値を指定する必要があります。 <ul style="list-style-type: none"> • 0 = すべて • 7 = 1 週間以内 • 30 = 1 ヶ月以内 • 180 = 6 ヶ月以内 • 365 = 1 年以内

登録チャンネルの使用

登録チャンネルには、カテゴリ登録、ディスカッション登録、および保存される検索などのタイプ別に登録が表示されます。登録のタイプごとに、次の情報が表示されます。

- 登録のラベル
- 特定の登録のヒット件数を示す、登録の詳細情報へのリンク

エンドユーザーは、登録チャンネルの「編集」ボタンを利用して、すべての登録の更新と登録解除を行えます。登録の一致するエンドユーザーへの通知は、登録チャンネルでグループ化されます。通知は、登録チャンネルの更新時に生成されます。管理者は、チャンネルの `refreshTime` プロパティを設定して、一定時間、コンテンツのキャッシュを実際に描画することができます。エンドユーザーが `refreshTime` パラメータより短い時間内に登録チャンネルのコンテンツを複数回更新しようとしても、コンテンツは実際のデータからは生成されず、キャッシュから読み取られます。`refreshTime` チャンネルのプロパティの値は、秒単位で設定できます。

ドキュメントが次の条件に一致すると、新規ドキュメントとしてエンドユーザーに通知されます。

- 登録カテゴリに分類され、検索対象 (scope) と経過時間 (lapsed time) の条件と一致する場合。
- 登録したディスカッションのコメントであり、検索対象 (scope) と経過時間 (lapsed time) の条件と一致する場合。
- 基本検索条件または詳細検索条件と一致し、経過時間 (lapsed time) の条件と一致する場合。

カテゴリに登録する

1. サンプルデスクトップにログインします。
カテゴリには、次の方法で登録できます。
 - カテゴリをブラウズする: これには「登録」リンクが含まれます。
 - カテゴリを示す検索結果: これには「登録」リンクが含まれます。
 - カテゴリ内の検索結果: これには「カテゴリに登録」リンクが含まれます。
2. 登録するカテゴリの隣にある登録用リンクを選択します。
登録情報を指定するページが表示されます。
3. 次の情報を指定します。
 - 登録の名前: カテゴリの名前
 - ターゲットカテゴリ: カテゴリの名前
 - 検索の範囲: 「検索」テキストフィールドと同様のクエリー文字列

- 検索対象期間: 指定したカテゴリへの登録対象となる期間。「すべて」、「1週間以内」、「1ヶ月以内」、「6ヶ月以内」、「1年以内」に設定できます。
4. 「完了」ボタンを選択します。
登録のリストにカテゴリが追加されます。

ディスカッションに登録する

1. サンプルデスクトップにログインします。
ディスカッションへの登録は、ディスカッションを表示するリンクから行えます。これには、「登録」リンクが含まれます。
2. 登録するディスカッションの「登録」リンクを選択します。
登録情報を指定するページが表示されます。
3. 次の情報を指定します。
 - 登録の名前: カテゴリの名前
 - ターゲットカテゴリ: カテゴリの名前
 - 検索の範囲: 「検索」テキストフィールドと同様のクエリー文字列
 - 検索対象期間: 指定したカテゴリに登録する期間。「すべて」、「1週間以内」、「1ヶ月以内」、「6ヶ月以内」、「1年以内」に設定できます。
 - 重要度: この重要度以上が登録の対象となるしきい値
4. 「完了」ボタンを選択します。
これで、ディスカッションに登録されます。

検索を保存する

1. サンプルデスクトップにログインします。
2. 「検索」タブを開き、ドキュメントを検索します。
検索結果のページが表示されます。
3. 結果リストの上に表示される登録のリンクをクリックします。
登録情報を指定するページが表示されます。
4. 次の情報を指定します。
 - ラベル: 保存される検索のラベル
 - 検索の範囲: 「検索」テキストフィールドと同様のクエリー文字列
 - 検索対象期間: 指定した検索結果に登録する期間。「すべて」、「1週間以内」、「1ヶ月以内」、「6ヶ月以内」、「1年以内」に設定できます。
5. 「完了」ボタンを選択します。

これで、検索結果が保存されます。

ディスカッション

ここでは、次の内容について説明します。

- [ディスカッションの概要](#)
- [DiscussionProvider](#)
- [チャンネルの管理と使用](#)

ディスカッションの概要

ディスカッションは、トピックと特定のドキュメントに関連付けられています。これは、既存のドキュメントを追加して、それについて話し合ったり、独自のトピックを作成する強力な手段です。特定のドキュメントや新しいトピックに関する情報を簡単に共有することができます。

Sun Java System Portal Server ソフトウェアのディスカッション機能には、ディスカッションスレッド、ドキュメントまたは新規トピックに基づくディスカッションの開始、ディスカッションの検索、ディスカッションの重要度設定が含まれます。デフォルトでは、匿名ユーザー用のサンプルポータルでも Discussions チャンネルを利用できます。ただし、匿名ユーザーはディスカッションに登録したり、Discussions チャンネルを編集したりすることはできません。

DiscussionLite チャンネルと Discussions チャンネルは DiscussionProvider に基づいています。検索チャンネルの JSP と同様に、クエリーの部分と表示の部分があり、デスクトップテーマを使用します。

DiscussionProvider

DiscussionProvider は、デスクトップテーマを使用する JSP プロバイダです。これは、検索タグライブラリと API を使用して、バックエンドの検索サービスからデータを取得します。ディスカッションとコメントは、異なるリソースディスクリプタ (RD) としてディスカッションデータベースに格納されます。DiscussionProvider がサポートする内容は次のとおりです。

- 次の機能を持つ完全ビュー (Discussions チャンネル経由) と省略ビュー (DiscussionLite チャンネル経由)
 - ディスカッションチャンネルから新しいディスカッションを開始する。

- 既存のディスカッションに返信する。
 - 検索チャンネルからの Web ドキュメントに基づく新しいディスカッションを開始する。
 - 次の機能を持つディスカッションリスト
 - 最終更新日でソートされた主要送信を取得する。
 - ユーザーが古いディスカッションにアクセスできるようにページ付けを行う。
 - ディスカッションの各サブツリーを表示するディスカッション表示。主要項目は詳細に表示され、サブツリーは主要項目の下に表示されます。ここに表示される内容には、次の項目が含まれます。
 - ページ上の複数のフィルタ。ドキュメントの重要度 (「重要でない (irrelevant)」、**「普通 (routine)」**、「興味 (interesting)」、「重要 (important)」、「必読 (must read)」) などのフィルタに基づいてドキュメントを表示できます。
 - スレッド表示とフラット表示の表示設定を選択できます。
 - 展開しきい値を使用すると、サブツリー内の表示を制御しやすくなります。ユーザーは、評価の高いドキュメントだけを展開したり、すべてのドキュメントを展開したり閉じたりすることを選択できます。デフォルト値は「すべて縮小」です。「すべて展開」を選択した場合、フィルタ条件を満たすすべてのコメントが展開されます。また、ディスカッションの説明およびディスカッション評価用メニューが表示されるほか、返信を投稿する仕組みも提供されます。
 - ディスカッション内の検索に対応しています。
- ユーザーは、チャンネル編集ページからこれらの設定を編集することもできます。
- ディスカッションに対してコメント追加し、重要度を設定する。たとえば、ユーザーは次の操作が可能です。
 - 既存のディスカッションにコメントを追加する。
 - すべてのディスカッションとコメントに重要度を設定する。ただし、ユーザーが設定する重要度は直ちには適用されません。重要度の計算は、コメントの重要度が段階的に上下するアルゴリズムに基づいています。たとえば、コメントの重要度を「重要」に引き上げるには、「重要」という設定を 3 回行う必要があります。
 - ディスカッション内のすべてのディスカッションを検索する。これらの機能は、検索プロバイダにルーティングされます。ユーザーは、詳細検索機能を使用して、重要度による検索も実行できます。
 - 登録。認証されたユーザーは、登録のリンクを選択することで、特定のディスカッションを選択的に登録できます。この要求は、SubscriptionProvider によって処理されます。この機能が必要ない場合は、displaySubscription プロパティを無効に設定できます (コード例 16-1 を参照)。このプロパティのデフォルト値は true です。

DiscussionProvider のディスプレイプロファイル XML コード

コード例 16-1 は、DiscussionProvider のディスプレイプロファイル XML コードを示しています。

コード例 16-1 DiscussionProvider プロバイダのディスプレイプロファイル XML コード

```
<Provider name="DiscussionProvider"
class="com.sun.portal.providers.jsp.JSPProvider">
  <Properties>
    <String name="title" value="*** Discussions Provider ***"/>
    <String name="description" value="*** DESCRIPTION ***"/>
    <String name="refreshTime" value="0" advanced="true"/>
    <String name="helpURL" value="en/desktop/discussions.htm"
advanced="true"/>
    <String name="fontFace1" value="Sans-serif"/>
    <String name="productName" value="Sun Java System Portal Server"/>
    <String name="contentPage" value="discussionContent.jsp"/>
    <String name="editPage" value="discussionEdit.jsp"/>
    <String name="processPage" value="discussionDoEdit.jsp"/>
    <Boolean name="isEditable" value="true" advanced="true"/>
    <String name="editType" value="edit_subset" advanced="true"/>
    <Boolean name="showExceptions" value="false"/>
    <Boolean name="showErrors" value="true"/>
    <String name="width" value="thick"/>
    <String name="column" value="2"/>
    <String name="searchServer" value=""/>
    <String name="dbname" value=""/>
    <Integer name="viewHits" value="8"/>
    <String name="defaultDiscussionDisplay" value="Threaded"/>
    <String name="defaultFilter" value="Irrelevant"/>
    <String name="defaultExpansionThreshold" value="Collapse all"/>
    <Boolean name="viewDiscussionWindow" value="false"/>
    <String name="anonymousAuthor" value="anonymous"/>
    <Boolean name="displaySearch" value="true"/>
    <Boolean name="showDescription" value="false"/>
    <String name="ratingText"
value="Irrelevant,Routine,Interesting,Important,Must Read"/>
  </Properties>
</Provider>
```

DiscussionProvider の管理

DiscussionProvider の管理は、次のように分散されています。

- チャンネル編集ページ (ユーザーによる設定が可能)
- 管理コンソールの DiscussionProvider のチャンネルデスクトップの「チャンネルおよびコンテンツを管理」リンク
- 検索サービス

DiscussionLite チャンネル

DiscussionLite チャンネルは、上位 20 のディスカッションタイトル (再設定可能) と日付を表示します。ディスカッションは、作成日 (最終更新日) でソートされ、最も新しいディスカッションが先頭に表示されます。DiscussionLite チャンネルの表示には、各ディスカッションを表示するリンク、Discussions チャンネルをターゲットとするすべてのディスカッションを表示するリンク、ディスカッションを開始するリンクがあります。デフォルトでは、チャンネルは単一のコンテンツ内に表示され、すべてのリンクは JSPDynamicSingleContainer に持ち込まれます。

プロパティは、管理コンソールから設定できます。デフォルトでは、このチャンネルのプロパティをユーザーが編集することはできません。

ディスカッションは、ディスプレイプロファイルの dbname プロパティで指定されるディスカッションデータベースに格納されます。ディスプレイプロファイルでは、検索サーバーホスト (searchServer プロパティ)、データベース名 (dbname プロパティ)、表示するディスカッションの数 (viewHits プロパティ) を設定できます (367 ページのコード例 16-2 を参照)。

コード例 16-2 DiscussionLiteProvider のディスプレイプロファイル XML コード

```
<Channel name="DiscussionLite" provider="DiscussionProvider">
  <Properties>
    <String name="title" value="Recent Discussions"/>
    <String name="description" value="This is a DiscussionLite provider example"/>
    <String name="contentPage" value="discussionLiteContent.jsp"/>
    <String name="editPage" value=""/>
    <String name="processPage" value=""/>
    <String name="width" value="thin"/>
    <String name="searchServer" value=""/>
    <String name="db" value="discussion"/>
    <Integer name="viewHits" value="20"/>
  </Properties>
</Channel>
```

DiscussionLite チャンネルでは、次の JSP が使用されます。

discussionLiteContent.jsp	JSP コンテンツページ。
query.jsp	検索クエリーを設定、実行する。
display.jsp	結果を表示する。
error.jsp	例外とエラーメッセージを表示する。

Discussions チャンネル

Discussions チャンネルによる完全な表示には、次の機能が含まれます。

- 降順にソートされた上位 8 ディスカッションの詳細な説明を表示する。これは、チャンネル編集ページで再設定できます。
- ユーザーがすべてのディスカッションを表示するためのページ付けを行う。
- 検索をサポートする。検索は、ディスカッションとコメントの結果を返します。

Discussions チャンネルのプロパティは、Sun Java System 管理コンソールから設定できます。

ディスカッションは、ディスプレイプロファイルの dbname プロパティで指定されるディスカッションデータベースに格納されます。ディスプレイプロファイルでは、検索サーバーホスト (searchServer プロパティ)、データベース名 (dbname プロパティ)、表示するディスカッションの数 (viewHits プロパティ) を設定できます ([コード例 16-3](#) を参照)。

コード例 16-3 Discussions チャンネルのディスプレイプロファイル XML コード

```
<Channel name="Discussions" provider="DiscussionProvider">
  <Properties>
    <String name="title" value="Discussions"/>
    <String name="description" value="This is a Discussion provider example"/>
    <String name="searchServer" value=""/>
    <String name="dbname" value="discussions"/>
    <Integer name="viewHits" value="8"/>
  </Properties>
</Channel>
```

Discussions チャンネルでは、次の JSP が使用されます。

<code>discussionContent.jsp</code>	JSP コンテンツページ
<code>discussionEdit.jsp</code>	編集ページ
<code>discussionDoEdit.jsp</code>	プロセス編集ページ
<code>declare.jsp</code>	
<code>portal.jsp</code>	ディスプレイプロフィールデータを抽出
<code>fullDiscussion.jsp</code>	完全な表示を処理
<code>fullDiscussionDisplay.jsp</code>	すべてのディスカッションページのユーザーインターフェース
<code>searchUI.jsp</code>	すべてのディスカッションページに表示される検索フォーム
<code>viewDiscussion.jsp</code>	ディスカッションを表示
<code>viewDiscussionBar.jsp</code>	ディスカッション表示ページ中央の水平バーで、すべてのフィルタが含まれる
<code>viewDiscussionDisplay.jsp</code>	ディスカッション表示ページのユーザーインターフェース
<code>viewDiscussionHeader.jsp</code>	ディスカッション表示ページに表示されるヘッダーコメント
<code>viewDiscussionNavigation.jsp</code>	ディスカッション表示ページのヘッダーの上下に表示されるナビゲーションバー
<code>feedback.jsp</code>	コメント、フィードバック、重要度の各機能を提供
<code>feedbackDisplay.jsp</code>	フィードバックを表示
<code>feedbackForm.jsp</code>	フィードバックフォームを提供
<code>feedbackProcess.jsp</code>	フィードバックを処理
<code>error.jsp</code>	例外とエラーメッセージを表示
<code>query.jsp</code>	検索クエリーの設定と実行
<code>pageFooter.jsp</code>	ページ付け

チャンネルの管理と使用

DiscussionProvider チャンネルの管理

DiscussionProvider チャンネルの管理は、デスクトップのディスプレイプロファイルと、Sun Java System Access Manager 管理コンソール上の検索サービスに分散して行われます。プロバイダに固有の情報は、ディスプレイプロファイルに格納されます。ディスカッションドキュメントとデータベースに関連する管理は、検索サービスで行う必要があります。

ディスカッションはディスカッションデータベースに格納されます。ディスカッションデータベースは、ディスカッションとコメントが特定のスキーマで構成されていることを前提としています。この機能のために、schema.rdm ファイルに新たにスキーマのフィールドが追加されました。データベースの管理とデバッグには、検索 CLI `rdmgr` を使用できます。たとえば、すべてのコメントをダンプするには、次のように入力します。

```
./run-cs-cli rdmgr -y discussion
```

DiscussionProvider チャンネルのサンプルは、デフォルトの検索サーバーを使用するように設定されています。ディスカッションデータベースとチャンネルにインポートされるいくつかのサンプルディスカッションは、すぐに使用できます。

サンプルが、`PortalServer-base/SUNWps/samples/discussions/` ディレクトリに格納されています。サンプルは次のとおりです。

<code>discussions.soif</code>	ディスカッションデータベースにロードされるサンプル SOIF ファイル。
<code>dp-org.xml</code>	ディスカッションチャンネルのディスプレイプロファイル XML コード。
<code>dp-providers.xml</code>	ディスカッションプロバイダのディスプレイプロファイル XML コード。
<code>dp-anon.xml</code>	認証なしの匿名ユーザー用の XML コード。サンプルポータルインストール時にロードされる。

管理者は、ディスカッションへのアクセスを制御できます (読み取り専用、または完全な非表示)。

DiscussionProvider からチャンネルを作成する

1. Sun Java System Access Manager 管理コンソールにログインし、「表示」プルダウンメニューから「サービス」を選択します。
左のパネルにサービスのリストが表示されます。
2. 「ポータルデスクトップ」、「チャンネルおよびコンテナを管理」の順に選択します。
右のパネルに「チャンネルおよびコンテナを管理」リンクが表示されます。
3. 「チャンネル」の下の「新規」ボタンを選択します。
追加するチャンネルのタイプを指定するページが表示されます。
4. 「チャンネル名」テキストボックスにチャンネルの名前を入力し、「プロバイダ」プルダウンメニューから「DiscussionProvider」を選択します。
5. 「了解」を選択します。
これにより、指定したプロバイダに基づいてチャンネルが作成されます。「キャンセル」ボタンをクリックすると、新しいチャンネルを作成せずに「チャンネルおよびコンテナを管理」ページに戻ります。
6. チャンネルテーブルで新たに作成したチャンネルの隣にある「編集」リンクをクリックします。
チャンネルのデフォルト値を設定するページが表示されます。
7. プロパティを編集し、「保存」ボタンを選択して編集内容を保存します。
次のディスプレイプロファイルプロパティは、このプロバイダに固有のプロパティです。

<code>searchServer</code>	検索サーバーへのパス。デフォルト値は <code>portal/search</code> です。
<code>dbname</code>	任意の有効なデータベース。
<code>viewHits</code>	表示するディスカッションの数。
<code>defaultDiscussionDisplay</code>	<code>flat</code> または <code>threaded</code> に設定し、コメントサブツリーの表示をフラットまたはスレッドに設定できます。
<code>defaultFilter</code>	ディスカッションの検索と表示に適用されるフィルタ。サブツリーの表示を制御します。「重要でない (<code>irrelevant</code>)」、「普通 (<code>routine</code>)」、「興味 (<code>interesting</code>)」、「重要 (<code>important</code>)」、「必読 (<code>must read</code>)」などの重要度に基づく設定が可能です。デフォルト値は「重要でない」なので、「重要でない」以上の重要度が指定されたすべてのコメントが表示されます。「必読」フィルタは、重要度が高く設定されたコメントを強調表示します。

defaultExpansionThreshold	「すべて展開」または「すべて縮小」に設定可能。デフォルト値は「すべて縮小」です。「すべて展開」に設定すると、フィルタリングされたすべてのコメントが展開され、説明、重要度メニューが表示されます。また、ユーザーはリンクをクリックして返信できます。
anonymousAuthor	
viewDiscussionWindow	
displaySearch	
showDescription	Discussions チャンネル用にこのプロパティを設定可能。
ratingText	ディスカッションのデフォルトの重要度として「重要でない」、「普通」、「興味」、「重要」、または「必読」を設定可能。

DiscussionProvider サンプルチャンネルの使用

新しいディスカッションを開始する

1. サンプルデスクトップにログインします。
2. 次のいずれかの方法で新しいディスカッションを開始します。
 - チャンネルから: 「コラボレーション」タブを選択し、「新しいディスカッションの開始」リンクを選択します。
 - 検索チャンネルから: ドキュメントの隣にある「新しいディスカッションの開始」を選択します。
3. 次の情報を指定します。
 - タイトル: ディスカッションのタイトル
 - メッセージ: ディスカッションの内容
 - 重要度: ディスカッションの重要度。「重要でない」、「普通」、「興味」、「重要」、「必読」のいずれか。
4. 「投稿」ボタンを選択します。

通信チャネルの設定

この章では、Sun™ Java System Portal Server の通信チャネルについて説明します。まず、一般的な情報を提供し、次に、インストールしてから設定を開始するまでの通信チャネルの状態について説明します。最後に、サイトの要件に合わせて通信チャネルを設定するための各種手順について説明します。

この章の大部分は設定に関する情報です。これには、管理者側とエンドユーザー側の設定が含まれます。エンドユーザーは、各チャネルの「編集」ボタンをクリックすることで、ポータルデスクトップから各チャネルの設定を直接編集できます。これにより、エンドユーザーは特定のサーバー設定情報を編集するための編集ページにアクセスし、アドレス帳チャネルに表示されるアドレス帳の数など、そのエンドユーザーがそのチャネルに表示できる固有の機能を編集することができます。

管理者は、エンドユーザーの編集オプションを制限または拡張することができます。管理者は、エンドユーザーによるサーバー設定の必要なしに、チャネルを事前に設定することができます。詳細については、[396 ページの「管理者のプロキシ認証: エンドユーザーによる信用情報の設定の禁止」](#)を参照してください。

管理者は各チャネルの編集ページを設計できるので、エンドユーザーが編集できる機能を選択することができます。詳細については、[391 ページの「アプリケーション設定の編集: 通信チャネル編集ページの設定」](#)を参照してください。

さらに、特定のアプリケーションのインスタンスが複数存在するサイトでは(たとえば、メールアプリケーションの 2 つ以上のインスタンス)、管理者はエンドユーザーがポータルデスクトップで第 2 のメールチャネルを設定できるようにすることができます。詳細については、[395 ページの「通信チャネルのタイプとしてエンドユーザーが複数インスタンスを設定できるようにする」](#)を参照してください。

この章で説明する内容は次のとおりです。

- [通信チャネルの概要](#)
- [通信チャネルでサポートされるソフトウェア](#)
- [インストーラと通信チャネル](#)
- [通信チャネルの設定タスク](#)

通信チャネルの概要

Sun Java System Portal Server 製品には、エンドユーザーがポータルデスクトップから直接アクセスできる 4 つの通信チャネルが用意されています。これらのチャネルを使用することで、エンドユーザーは対応するアプリケーション (メールアプリケーション など) にアクセスし、効果的かつ効率的に調整、計画、通信することができます。

4 つの通信チャネルは次のとおりです。

アドレス帳チャネル : アドレス帳チャネルには、エンドユーザーが表示できるアドレス帳エントリが表示されます。アドレス帳にアクセスしてアドレス帳エントリを作成、編集するには、「アドレス帳の起動」をクリックします。

カレンダーチャネル : カレンダーチャネルには、エンドユーザーが表示できるカレンダーイベントが表示されます。カレンダーアプリケーションにアクセスして新しいタスクやイベントを作成するには、「カレンダーを起動」をクリックします。

Instant Messaging チャネル : Instant Messaging チャネルには、Sun™ Java System Instant Messenger にアクセスできる他のユーザーのアクセス状態が表示されます。これらの連絡先は、Instant Messenger アプリケーションを使用してエンドユーザーが作成したリストから取得されます。アクセス中の状態を示すアイコンをクリックすることで、チャネルからのチャットを開始できます。これは、Instant Messenger を起動する方法の一つです。アクセス状態をチャネルから直接更新するには、ポータルデスクトップを再読み込みします。状態が変更するたびにアクセス状態を更新するには、「Instant Messenger」をクリックして Instant Messenger を起動し、連絡先のアクセス状態を表示します。

メールチャネル : メールチャネルには、エンドユーザーが表示できる、そのエンドユーザー宛てに送信されたメールメッセージが表示されます。メールアプリケーションにアクセスしてメッセージを表示、作成するには、「メールを起動」をクリックします。

通信チャネルでサポートされるソフトウェア

Sun Java System Portal Server ソフトウェアは、通信チャネルで次のリソースサーバープラットフォームをサポートします。

- Sun™ Java System Messaging Server 5.2、6.0、6 2004Q2
- Sun™ Java System Calendar Server 5.1.1、6.0、6 2004Q2
- Sun™ Java System Instant Messaging Server 6.1、6 2004Q2
- IBM Lotus Notes 5.0.6
- Microsoft Exchange Server 2000

インストーラと通信チャンネル

Sun Java System Portal Server のインストーラは、通信チャンネルに関連していくつかのタスクを実行します。通信チャンネルの一般的な設定タスクもインストーラによって行われます。管理者とエンドユーザーは、サイトの要件や、各自の必要に合わせてさらに詳細な設定を行う必要があります。

Sun Java System Portal Server インストーラが実行するタスク

Sun Java System Portal Server のインストーラは、次のタスクを実行します。

- SUNWpsso、SUNWpsap、SUNWpsmp、SUNWpscp、および SUNWiimps の各パッケージをインストールする。これらのパッケージはデフォルトの Sun Java System Portal Server インスタンスに配備されます。つまり、インストーラは Sun Java System Portal Server のすべてのインスタンスに通信チャンネルをインストールするわけではありません。複数サーバーでの配備については、[376 ページの「複数インスタンスでの配備」](#)を参照してください。
- アドレス帳、カレンダー、インスタントメッセージ、メールのチャンネルを作成します。インストーラは、Sun Java System サーバーのチャンネルをサンプル組織の「フロントページ」タブパネルコンテナに配置します。つまり、サンプルポータルをインストールした場合にだけ通信チャンネルはインストールされます。Microsoft Exchange Server と IBM Lotus Notes サーバーは、自動的にコンテナに配置されません。管理者は、必要に応じてこれらのチャンネルを追加する必要があります。

カレンダーチャンネルとメールチャンネルのデフォルトの設定は、エンドユーザーが基本的な設定を行うまで機能しません。このため、管理者はこれらのチャンネルをさらに設定する必要はありません。アドレス帳チャンネルと Instant Messaging チャンネルは、管理者とエンドユーザーによる追加設定を必要とします。

- Sun Java SystemCalendar Server および Sun Java System Messaging Server とのシングルサインオンを有効にするためのシングルサインオン (SSO) アダプタサービスを作成、設定します。

複数インスタンスでの配備

複数の Sun Java System Portal Server が配備されている場合は、Sun Java System Portal Server の各追加インスタンスに通信チャンネルを手動で配備し、各インスタンスを再起動する必要があります。この配備を行うには、次のように入力します。

```
PortalServer-base/SUNWps/bin/deploy redeploy -instance instancename  
-deploy_admin_password deployadminpassword
```

この *instancename* はデフォルト以外の特定のインスタンスの名前で、*deployadminpassword* は Web コンテナ (Web サーバーまたはアプリケーションサーバー) の管理者パスワードです。Web コンテナの管理者パスワードは、Web コンテナが Sun™ Java System Application Server または BEA WebLogic Server™ である場合にだけ必要となります。パスワードを含めておくことで、それを受け入れるその他の Web コンテナを使用するときに手間を省けます。Sun™ Java System ただし、Web Server または IBM WebSphere® Application Server では、このような場合パスワードは無視されます。

コード例 17-1 は、Sun Java System Portal Server のデフォルト以外の 2 つのインスタンスに通信チャンネルを手動で配備し、それぞれのインスタンスを再起動するコマンドを示しています。myinstance1 と myinstance2 は Sun Java System Portal Server のデフォルト以外のインスタンスの名前、Admin は Web コンテナの管理者パスワードです。

コード例 17-1 デフォルト以外のインスタンスへの通信チャンネルの配備

```
portalServer-base/SUNWps/bin/deploy redeploy -instance myinstance1  
-deploy_admin_password Admin  
portalServer-base/SUNWps/bin/deploy redeploy -instance myinstance2  
-deploy_admin_password Admin
```

通信チャネルの設定タスク

次に、通信チャネルを設定する手順の概要を示します。すべてのタスクがすべてのサイトに適用されるわけではありません。サイトのビジネス要件に基づいて、サイトに適用されるタスクを決定する必要があります。

- メールアプリケーションとカレンダーアプリケーションへのアクセスを有効化する
- デフォルトの組織のサービスを設定する
- エンドユーザーとしてチャネルを設定する
- アプリケーション設定の編集: 通信チャネル編集ページの設定
- 通信チャネルのタイプとしてエンドユーザーが複数インスタンスを設定できるようにする
- 管理者のプロキシ認証: エンドユーザーによる信用情報の設定の禁止
- 認証不要ポータルデスクトップ用に読み取り専用通信チャネルを設定する
- Microsoft Exchange Server または IBM Lotus Notes の設定
- デフォルト組織の下に新規ユーザーを作成する
- HTTPS が有効な Messaging Server とメールプロバイダが連携するように設定する

同じサーバー、または異なるサーバーに Sun Java System Messaging Server と Sun Java System Calendar Server がすでにインストールされている場合は、チャネル作成時に対応する URL を指定してください。

メールアプリケーションとカレンダーアプリケーションへのアクセスを有効化する

Messaging Server および Calendar Server 両方が、ブラウザによりログインセッション ID が要求されるホストの、インターネットプロトコル (IP) アドレスを確認します。IP アドレスが、セッション ID が発行されているホストのホスト IP アドレスと異なる場合、Messaging Server および Calendar Server は、セッションのタイムアウトメッセージを出して、セッションを拒否します。

IP セキュリティチェックを有効または無効にするパラメータの値を変更して、ユーザーが Portal Server を介してメールにアクセスできるようにする必要があります。ログイン IP アドレスへのセッションのアクセスを制限するかどうかを指定するパラメータは、次の通りです。

```
service.http.ipsecurity
```

Messaging Server 用の ipsecurity を無効にする

Messaging Server 用の ipsecurity を無効にするには、メールサーバーを実行するコンピュータのコマンド行で、次の手順を実行します。

1. Messaging Server にログインします。

2. 次のコマンドを入力します。

```
MessagingServer-base/sbin/server5/msg-messaging-server-hostname/configutil -o  
service.http.ipsecurity -v no
```

3. `su root` コマンドを使って `root` に変更します。

4. 次のコマンドを使って、Messaging Server を停止します。

```
MessagingServer-base/sbin/server5/msg-messaging-server-hostname/stop-msg
```

5. 次のコマンドを使って、Messaging Server を起動します。

```
MessagingServer-base/sbin/server5/msg-messaging-server-hostname/start-msg
```

Calendar Server 用の ipsecurity を無効にする

Calendar Server 用の ipsecurity を無効にするには、Calendar Server を実行するコンピュータのコマンド行で、次の手順を実行します。

1. Calendar Server にログインします。

2. Calendar Server が `/opt/SUNWics5` にインストールされていると仮定する場合、次を入力します。

```
cd /opt/SUNWics5/cal/config/
```

3. `ics.conf` ファイルを編集して、`ipsecurity` を `no` に設定します。例は次のとおりです。

```
service.http.ipsecurity = "no"
```

4. Calendar Server が `/opt/SUNWics5` にインストールされていると仮定する場合、次を入力して Calendar Server を再起動します。

```
/opt/SUNWics5/cal/sbin/stop-cal
```

```
/opt/SUNWics5/cal/sbin/start-cal
```

ポータルデスクトップを更新または再認証します。「カレンダーを起動」リンクが機能します。

デフォルトの組織のサービスを設定する

通信チャネルのインストールが完了した後で、後述する方法で **Instant Messaging** チャネルとアドレス帳チャネルをより詳細に設定する必要があります。カレンダーチャネルとメールチャネルには、デフォルトの設定が適用されるので、管理者による追加設定なしで利用できます。カレンダーチャネルとメールチャネルを含め、通信チャネルにサイト固有の問題が存在する場合がありますので注意してください。問題を解決するために、管理者がサイトの要件に合わせてチャネルを設定しなければならないこともあります。

次に、通信チャネルの設定に関連する重要な情報を示します。

[通信チャネルの設定に関する情報](#)

[Instant Messaging チャネルの設定](#)

[アドレス帳チャネルの設定](#)

通信チャネルの設定に関する情報

すべての通信チャネルについて

エンドユーザーの設定

通信チャネルにプロキシ認証を設定する場合を除き (詳細については [396 ページの「管理者のプロキシ認証: エンドユーザーによる信用情報の設定の禁止」](#) を参照)、エンドユーザーは、各通信チャネルの「編集」ボタンをクリックしてチャネルの編集ページにアクセスし、チャネルをさらに設定する必要があります。

注意: 検出されないエラー: 起動リンクが表示されない

いずれかの通信チャネルでクライアントのポート番号を誤って入力しても、エンドユーザーにはエラーメッセージが表示されません。該当するチャネルの起動リンクが表示されないため、エラーがあること自体は特定できますが、この情報だけでエンドユーザーが問題の根本的な原因を特定するのは困難です。管理者とエンドユーザーの両方が誤ったクライアントポート番号を入力する可能性があります。カレンダーチャネルとメールチャネルについてはエンドユーザーだけがクライアントポート番号を編集できるため、これらのチャネルでこの問題が生じた場合は、エンドユーザーによる誤入力の原因であると言えます。

注意: 検出されないエラー: チャネルが表示されない

エンドユーザーに通信チャネルが表示されず、エラーメッセージも表示されないさまざまな状況があります。テンプレートまたは設定の名前を誤って設定した可能性があるため、テンプレートや設定を見つけることができないのです。次の条件のいずれかを満たすときには、通信チャネルが表示されません。

- SSO アダプタテンプレートが見つからない場合。
- SSO アダプタ設定が見つからない場合。

- `display.template` ファイルが見つからない場合。

メールチャネルについて

HTTPS が有効な Messaging Server

メールチャネルが通常の HTTP メッセージングサーバーではなく、より安全な HTTPS が有効なメッセージングサーバーに接続される場合は、メールチャネルが正常に機能するように、セキュリティに関連する調整を行う必要があります。詳細については、[419 ページの「HTTPS が有効な Messaging Server とメールプロバイダが連携するように設定する」](#)を参照してください。

Instant Messaging チャネルの設定

Sun Java System Sun Java System Instant Messaging Server のインストール時に、Portal Server の「IM を有効にする」オプションを選択した場合は、Sun Java System Portal Server のインストール時に Instant Messaging Services がインストールされません。

インスタントメッセージポータルチャネルはそのままでも機能するように設計されていますが、サイトの要件に合わせて設定が必要となることもあります。インストールに追加の設定が必要であるかどうかを調べるには、「[Instant Messaging チャネルを設定する](#)」の手順を実行した後に、「[Instant Messaging チャネルの追加設定](#)」の各節を参照してください。

Instant Messaging チャネルは、IMProvider という Portal Server コンテンツプロバイダに基づいています。IMProvider は、Portal Server 内の JSPProvider の機能拡張です。JSPProvider の機能拡張であるため、IMProvider は Instant Messaging チャネルのコンテンツページと編集ページの生成に JSP ファイルを使用します。Instant Messenger を起動するためのページも JSP ファイルを使用して生成されます。IMProvider は、インスタントメッセージに固有のタグライブラリも定義します。このタグライブラリは、JSP ファイルによって使用されます。JSP ファイルとタグライブラリは、IMProvider によって定義されるチャネルプロパティを使用します。

Sun Java System Instant Messaging Server の詳細については、『[Instant Messaging 管理者ガイド](#)』を参照してください。Sun Java System Portal Server の Instant Messaging チャネルタグライブラリ、および JSP ファイルの編集による Instant Messaging チャネルのカスタマイズについては、『[Sun Java System Portal Server 6 2005Q1 Desktop Customization Guide](#)』を参照してください。また、管理者とエンドユーザーは、Instant Messaging チャネル設定のコードベースプロパティで使用されている URL にアクセスすることで、Sun Java System Instant Messaging Server に関する情報を参照できます。

Instant Messaging チャネルを設定する

1. インターネットブラウザで `http://hostname:port/amconsole` (たとえば `http://psserver.company22.example.com:80/amconsole`) と入力し、Sun™ Java System Access Manager の管理コンソールにログインします。
2. 「アイデンティティ管理」タブをクリックし、ナビゲーションパネル(左下のフレーム)に「表示」ドロップダウンリストを表示します。
3. 「表示」ドロップダウンリストから「サービス」を選択し、設定可能なサービスのリストを表示します。
4. 「Portal Server 設定」の見出しの下で、「ポータルデスクトップ」の隣にある矢印をクリックします。データパネル(右下のフレーム)に「ポータルデスクトップ」ページが表示されます。
5. 「チャネルおよびコンテナを管理」をクリックします。
6. 「チャネル」という見出しまでスクロールし、「IMChannel」の隣にある「プロパティの編集」をクリックして Instant Messaging サービスのパネルを表示します。このパネルには、「基本プロパティ」があります。

次に、IMChannel の編集ページに表示されるプロパティの一部を示します。各プロパティには値の例が指定されています。

プロパティ	値の例
authMethod	idsvr
authUsernameAttr	uid
clientRunMode	plugin
codebase	imapplet.example.com
contactGroup	My Contacts
mux	imserver.example.com
muxport	49909
netletRule	IM
password	(authmethod が idsvr の場合は適用されない)
port	49999
server	imserver.example.com
username	(authmethod が idsvr の場合は適用されない)

7. 設定する各プロパティの隣にあるテキストフィールドに適切な値を入力します。
表 17-1 は、プロパティの説明と、値として入力できる情報のタイプを示しています。

表 17-1 IMChannel の編集ページのプロパティと値の説明

プロパティ	値
authMenthod	通常は、認証方法として、Sun Java System Access Manager の認証方法が使用されることを示す <code>idsvr</code> を、値として入力するようにします。 指定できる値は、 <code>idsvr</code> または <code>ldap</code> の 2 つです。 <code>idsvr</code> 値を指定すると、シングルサインオンが有効になります。また、 <code>username</code> フィールドと <code>password</code> フィールドが Instant Messaging チャンネル編集ページから取り除かれます。
authUsernameAttr	認証方法 <code>idsvr</code> による認証では、ユーザー名用に使用される属性の名前を入力します。
clientRunMode	インスタントメッセージクライアントを実行する方法を入力します。指定できる値は <code>plugin</code> または <code>jnlp</code> です (Java Web Start で使用)。
codebase	インスタントメッセージクライアントがダウンロードされる URL のプレフィックスを入力します。
contactGroup	Instant Messaging チャンネルで表示される連絡先グループの名前を入力します。
mux	チャンネルがインスタントメッセージクライアントを起動したときに使用される Sun Java System Instant Messaging Multiplexor のホスト名を入力します。
muxport	Sun Java System Instant Messaging Multiplexor に関連付けられているポート番号を入力します。デフォルトのポート番号は 49909 です。
netletRule	Secure Remote Access (SRA) ゲートウェイ経由のセキュアモードの場合は、インスタントメッセージクライアントが使用する Netlet ルールの名前を入力します。
password	認証方法 LDAP による認証で使用されるパスワードを入力します。このプロパティは、ディスプレイプロファイルに格納されるときに、 <code>AMPasswordUtil</code> クラスを使用して難読化されます。
port	チャンネルが使用する Sun Java System Instant Messaging Server が関連付けられているポート番号を入力します。デフォルトのポート番号は 49999 です。

表 17-1 IMChannel の編集ページのプロパティと値の説明 (続き)

プロパティ	値
server	チャネルが使用する Sun Java System Instant Messaging Server のホスト名を入力します。
username	認証方法 LDAP による認証で使用されるユーザー名を入力します。

8. 必要に応じてページをスクロールし、「保存」をクリックします。

Instant Messaging チャネルの追加設定

複数の組織への対応には追加手順の実行が必要になることがある

Portal Server インスタンスが 1 つの Instant Messaging サーバーを使用して複数の組織にサービスを提供する場合は、追加手順を実行する必要があります。

Access Manager と Portal Server では、管理者は複数のユーザーが組織内で同じユーザー ID (uid) を持つように設定することができます。たとえば、組織に 2 つのサブ組織があり、それぞれが enduser22 という名前のエンドユーザーを持つことができます。このとき、この 2 人が Instant Messaging チャネルを通じてそれぞれのアカウントにアクセスしようとする、競合が発生します。

このような競合の可能性をなくすために、組織ごとに JSP 起動ページのセットを作成し、組織の sunPreferredDomain 属性の値に設定されたパラメータを含め、この値をドメインとして渡す必要があります。デフォルトの起動ページは次のとおりです。

```
/etc/opt/SUNWps/desktop/default/IMProvider/jnlpLaunch.jsp
```

```
/etc/opt/SUNWps/desktop/default/IMProvider/pluginLaunch.jsp
```

組織への Instant Messenger リンクの追加

デフォルトでは、デフォルト組織のアプリケーションチャネルに Instant Messenger リンクが追加されます。このアプリケーションチャネルには、さまざまなアプリケーションを起動するためのリンクがあります。Instant Messenger リンクを使用することで、アプリケーションチャネルから Instant Messenger を起動できます。次の場合は、Instant Messenger リンクを手動で追加する必要があります。

- 別の組織にこれらのリンクを追加する場合。
- サンプルポータルがインストールされていない場合。
- 別のチャネルで AppProvider を使用している場合。

Instant Messenger リンクのコンテンツは

PortalServer-base/SUNWps/samples/InstantMessaging/dp-IMChannel.xml ファイルに記録されています。dp-IMChannel.xml ファイルには、サンプルの IMChannel も含まれません。

別の組織のディスプレイプロファイルに Instant Messenger リンクの情報を追加するには、次のように dp-IMChannel.xml ファイルのコピーを編集し、dpadmin コマンドを実行してそのファイルをインストールします。

1. 次のディレクトリに移動します。

```
PortalServer-base/SUNWps/bin/
```

2. 次の方法で dp-IMChannel.xml ファイルのコピーを作成します。

```
cp dp-IMChannel.xml newfile.xml
```

3. 次の dpadmin コマンドを実行して、アプリケーションチャネルを修正します。

```
dpadmin modify -u ADMIN_DN -w PASSPHRASE -d ORG_DN -m  
newfile.xml
```

各表記の意味は次のとおりです。

ADMIN_DN は LDAP 管理者の DN です。たとえば、次のようになります。
amadmin

PASSPHRASE は管理者のパスワードです。

ORG_DN はリンクを追加する組織の DN です。たとえば、次のようになります。
o=example.com, o=isp

Instant Messaging チャネルは、起動引数として Java プラグインを使用して Instant Messenger を起動するための URL を参照します。たとえば、次のようになります。

```
/portal/dt?action=content&provider=IMChannel&launch=plugin&username=sam
```

Java Web Start を使用して Instant Messenger アプレットを起動するための URL は、次のようになります。

```
/portal/imlaunch?channel=IMChannel&launch=jnlp&username=sam
```

Sun Java Server Portal Server での Sun Java System Instant Messenger のセキュアモードの有効化

Netlet は、Instant Messenger とサーバーの間にセキュリティ保護された通信を提供します。

注 Secure Remote Access ゲートウェイ経由でアクセスした場合、Instant Messaging チャネルは自動的にセキュアモードを使用します。ゲートウェイを経由せずにアクセスした場合は、Instant Messaging チャネルはセキュアモードを使用しません。

セキュアモードを有効にするには、Netlet ルールを追加する必要があります。

Netlet ルールを追加するには、次の手順を実行します。

1. インターネットブラウザで `http://hostname:port/amconsole` (たとえば `http://psserver.company22.example.com:80/amconsole`) と入力し、Access Manager の管理コンソールにログインします。
2. 「アイデンティティ管理」タブをクリックし、ナビゲーションパネルに「表示」ドロップダウンリストを表示します。
3. 「表示」ドロップダウンリストから「サービス」を選択し、設定可能なサービスのリストを表示します。
4. 「SRA 設定」までスクロールし、「Netlet」を選択します。
5. 「Netlet」の隣にある矢印のアイコンをクリックします。右のパネルに「Netlet ルール」が表示されます。
6. 「Netlet ルール」横の「追加」をクリックします。
7. 「ルール名」フィールドに IM と入力します。

注 別の Netlet ルール名を指定することもできます。別の Netlet ルールを使用するように Instant Messaging チャネルを設定できます。

8. URL フィールドのデフォルト値を消去し、空白のまま残します。
9. 「アプレットのダウンロード」チェックボックスにチェックマークを付けて、次の文字列を入力します。

`$IM_DOWNLOAD_PORT:$IM_HOST:$IM_PORT`

たとえば、次のようになります。

`49916:company22.example.com:80`

各表記の意味は次のとおりです。

IM_DOWNLOAD_PORT: Netlet を使用してインスタントメッセージリソースがダウンロードされるポート。

IM_HOST: Instant Messenger にサービスを提供する Web コンテナのホスト名。たとえば、次のようになります。company22.example.com

IM_PORT: Instant Messenger にサービスを提供する Web コンテナのポート番号。たとえば、80。

10. 「ポート - ホスト - ポート」リストのデフォルト値を選択し、「削除」をクリックします。
11. 「クライアントポート」フィールドに、Netlet が実行されるローカルホストのポートを入力します。たとえば、次のようになります。49916
12. 「ターゲットホスト」フィールドに、インスタントメッセージマルチプレクサのホスト名を入力します。
13. 「ターゲットポート」フィールドに、インスタントメッセージマルチプレクサのポート番号を入力します。

注 Netlet のポート、Instant Messenger のホスト、Instant Messenger のポートは、インスタントメッセージサービスパネルで指定した Instant Messaging サービスの属性と一致させる必要があります。これについては、[381 ページの「Instant Messaging チャネルを設定する」](#)の最後の手順で説明しています。

14. 「リストに追加」をクリックします。
15. 「保存」をクリックして Netlet ルールを保存します。

Instant Messenger のユーザーによる起動の禁止

ユーザーのディスプレイプロファイルから Instant Messaging チャネルを消去し、ユーザーがこのチャネルを使用できないように設定することができます。たとえば、自動的にインストールされるサンプルの IMChannel を消去するには、次の手順を実行します。

1. インターネットブラウザで `http://hostname:port/amconsole` (たとえば `http://psserver.company22.example.com:80/amconsole`) と入力し、Access Manager の管理コンソールにログインします。
2. 「アイデンティティ管理」タブをクリックし、ナビゲーションパネルに「表示」ドロップダウンリストを表示します。
3. 「表示」ドロップダウンリストから「サービス」を選択し、設定可能なサービスのリストを表示します。
4. ポータルデスクトップサービスの隣にある矢印をクリックします。

5. 「チャネルおよびコンテナを管理」リンクをクリックします。
6. IMChannel チャネルの左のチェックボックスにチェックマークを付けます。
7. 必要に応じてページをスクロールし、「削除」をクリックしてチャネルを削除します。

アドレス帳チャネルの設定

アドレス帳チャネルを機能させるには、アドレス帳チャネルサービスのデフォルトを設定する必要があります。AddressBookProvider は事前に設定されていないため、AddressBookProvider に基づいて作成されるチャネルは、AddressBookProvider を設定するまでユーザーのデスクトップまたはコンテンツリンクに表示されません。

注 あらかじめ生成されているほかの通信チャネルに基づいてユーザー定義のチャネル設定を作成すると、メッセージ「有効な設定を指定してください。」が表示されることがあります。ほかの通信チャネルがユーザーのデスクトップに表示されるように定義されている場合でも、使用するバックエンドサービスを確定するために追加の管理作業を行う必要があります。

また、管理作業が完了した後に、デスクトップユーザーはバックエンド信用情報(ユーザー名とパスワードなど)を通信チャネルに指定する必要があります。デスクトップユーザーは、そのチャネルの「編集」ボタンを使用してこれらの値を指定できます。

注 userDefinedChannels 設定には、特定のセットアップに対して適用されないバックエンドサービスへの参照が含まれるため、この設定はインストールごとの管理が必要になる場合があります。たとえば、この設定におけるすべての Lotus プロバイダが、ポータルユーザーベースに Lotus バックエンドサービスを使用するユーザーがいる場合にのみ適用される通信チャネルの Lotus バックエンドサービスとの操作を参照します。

アドレス帳サービスのデフォルトの設定

ここでは、シングルサインオン (SSO) アダプタテンプレートについて説明します。これらのテンプレートは、ユーザーのポータルデスクトップでの通信チャネルの表示にグローバルに影響します。通信チャネル用にユーザーのディスプレイプロファイルを修正するには、SSO アダプタのテンプレートと設定を編集または作成する必要があります。

この章では、アドレス帳のテンプレートについてだけ説明します。アドレス帳についての説明も、ここではかなり限定的です。SSO アダプタ、SSO アダプタテンプレート、SSO アダプタの設定に関する広範な解説については、[453 ページの付録 A 「SSO アダプタテンプレートと設定」](#)を参照してください。

アドレス帳サービスのデフォルトを設定する

1. インターネットブラウザで `http://hostname:port/amconsole` (たとえば `http://psserver.company22.example.com:80/amconsole`) と入力し、Access Manager の管理コンソールにログインします。
2. 「サービス設定」タブをクリックします。設定できるサービスのリストがナビゲーションパネルに表示されます。
3. 「シングルサインオンアダプタの設定」という見出しが表示されるまでナビゲーションパネルをスクロールし、SSO アダプタ項目の隣にある矢印をクリックします。データパネルに「SSO アダプタ」ページが表示されます。
4. 「SSO アダプタの設定」から「新規」をクリックして、SSO アダプタの設定を追加します。
「新規設定」ページが表示されます。
5. 設定名を入力し、メニューから「SUN-ONE-ADDRESS-BOOK」を選択します。
6. 「次へ」をクリックします。
「設定プロパティ」ページが表示されます。
7. 必要に応じてプロパティを修正します。
8. 「SSO アダプタ」ページを下にスクロールし、「保存」をクリックします。
9. 完了したら、「保存」を選択します。

SSO アダプタテンプレート文字列の属性については、[453 ページの付録 A 「SSO アダプタテンプレートと設定」](#)を参照してください。

エンドユーザーとしてチャネルを設定する

1. 新規ユーザーとしてデスクトップにログインします。
 - a. インターネットブラウザで、次の URL にアクセスします。
`http://hostname.domain:port/portal/dt`、たとえば、
`http://psserver.company22.example.com:80/portal/dt`
 - b. ユーザー ID とパスワードを入力します。
 - c. 「ログイン」をクリックします。
2. 各チャネルの「編集」ボタンをクリックし、サーバー設定を編集します。
 - o. メールチャネルを設定するには、次の手順に従います。

サーバー名：メールサーバーのホスト名を入力します。たとえば、`mailserver.example.com` のように指定します。

IMAP サーバーポート：メールサーバーのポート番号を入力します。

SMTP サーバー名：送信メールサーバー (SMTP (Simple Mail Transfer Protocol) サーバー) のドメイン名サーバー (DNS) 名を入力します。

クライアントポート：HTTP サービスに設定されているポート番号を入力します。

ユーザー名：メールサーバーのユーザー名を入力します。

ユーザーパスワード：メールサーバーのユーザーパスワードを入力します。

メッセージ送信時に送信済みフォルダにコピーする：送信メッセージのコピーを「送信済み」フォルダに格納するときは、このチェックボックスにチェックマークを付けます。

完了：メール設定を保存するときは、このボタンをクリックします。

取消し：詳細な設定を保存せずにウィンドウを閉じるときは、このボタンをクリックします。

- アドレス帳チャネルを設定するには、次の手順を実行します。

IMAP のユーザー ID とパスワードは、メールチャネルの設定時に入力した「ユーザー名」、「ユーザーパスワード」と同じです。詳細については、「[メールチャネルを設定するには、次の手順に従います。](#)」の項目を参照してください。

ユーザー名：ユーザー名を入力します。

ユーザーパスワード：パスワードを入力します。

完了：サーバー情報を保存するときは、このボタンをクリックします。

取消し：詳細な設定を保存せずにウィンドウを閉じるときは、このボタンをクリックします。

- カレンダーチャネルを設定するには、次の手順を実行します。

サーバー名：カレンダーサーバーのホスト名を入力します。たとえば、`Calserver.example.com` のように指定します。

サーバーポート：カレンダーサーバーのポート番号を入力します。

ユーザー名：カレンダーサーバーのユーザー名を入力します。

ユーザーパスワード：カレンダーサーバーのユーザーパスワードを入力します。

完了：カレンダー設定を保存するときは、このボタンをクリックします。

取消し：詳細な設定を保存せずにウィンドウを閉じるときは、このボタンをクリックします。

- Instant Messaging チャネルを設定するには、次の手順に従います。

連絡先一覧 : Instant Messaging チャネルに表示する連絡先のリストを選択します。

起動方法 : 適切な起動方法を選択します。Java プラグインまたは Java Web Start を選択できます。

サーバー : Sun Java System Instant Messaging Server 名を入力します。たとえば、次のようになります。
IMserver.example.com

サーバーポート : Sun Java System Instant Messaging Server のポート番号を入力します。たとえば、次のようになります。
49999

マルチプレクサ : Sun Java System Instant Messaging サーバーと同じコンピュータでなければなりません。たとえば、次のようになります。
IMserver.example.com

マルチプレクサポート : マルチプレクサのポート番号を入力します。たとえば、次のようになります。
49909

ユーザー名 : (認証方法が、Sun Java System Access Manager 認証方法の idsvr に設定されている場合のみこのフィールドが表示される) Sun Java System Instant Messaging ユーザー名を入力します。

ユーザーパスワード : (認証方法が、Sun Java System Access Manager 認証方法の idsvr に設定されている場合のみこのフィールドが表示される) Sun Java System Instant Messaging ユーザーパスワードを入力します。

完了 : Sun Java System Instant Messaging Server 設定を保存するときは、このボタンをクリックします。

取消し : 詳細な設定を保存せずにウィンドウを閉じるときは、このボタンをクリックします。

注 アドレス帳、カレンダー、メールチャネルには、それぞれに表示オプションがあり、ユーザーがそれを設定できます。デフォルトでは、管理者はこの設定を変更しません。ユーザーがチャネルの表示オプションを変更するには、ポータルデスクトップにログインし、各チャネルのパネルで「編集」ボタンをクリックします。表示オプションが表示され、簡単に変更できます。

アドレス帳では、ユーザーは「エントリ数」表示オプションを変更できます。カレンダーでは「日」表示、メールでは「ヘッダー数」の各表示オプションをユーザーが変更できます。

デフォルトの通信チャネル表示オプションにユーザーが加えた変更は、優先して適用されます。後から管理者が加えた変更は自動的に有効にならず、管理者が追加した新規チャネルにもユーザーは自動的にアクセスできません。

アプリケーション設定の編集：通信チャネル編集ページの設定

エンドユーザーがアドレス帳、カレンダー、メールの各通信チャネルのツールバーで「編集」ボタンをクリックしたときに表示される編集ページを設定することができます。Instant Messaging チャネルの編集には、アプリケーション設定の編集は使用されません。Instant Messaging チャネルの編集ページの設定については、『Sun Java System Portal Server 6 2005Q1 Desktop Customization Guide』を参照してください。

アプリケーション設定の編集に対応している3つの通信チャネルでは、エンドユーザーにどのオプションの編集を許可するか、各オプションの名称と説明、オプションの形式を変更できます。通信チャネルの編集ページの設定には、ディスプレイプロファイル、各種HTMLテンプレート、SSOアダプタテンプレートを使用します。また、SSOアダプタ設定へのアクセスが必要になることもあります。編集ページの設定には、これらの項目すべてが関連します。

ここでは、アプリケーション設定の編集について概要だけを説明します。テンプレートファイルとディスプレイプロファイルの詳細な説明(相互動作の仕組みやアクセス方法、編集方法を含む)については、本書の各章と『Sun Java System Portal Server 6 2005Q1 Desktop Customization Guide』を参照してください。

編集ページのディスプレイプロファイルの属性

通信チャネルのディスプレイプロファイルには、編集ページの作成を指示する `ssoEditAttributes` および `dpEditAttributes` という2つのコレクションがあります。

Sun Java System Access Manager の管理コンソールにアクセスすることで、これらのコレクションを編集できます。ディスプレイプロファイルをダウンロードし、XML コードを編集して Directory Server にアップロードするか、管理コンソールだけを使用して、これらのコレクションの特定のプロパティを編集します。

ssoEditAttributes コレクションは、SSO アダプタサービスに含まれる属性 (user name、user password など) の編集を制御し、dpEditAttributes コレクションは、デフォルトではエンドユーザーに編集が許可されているディスプレイプロファイル属性 (sort order、sort by など) の編集を制御します。

このため、これらのコレクションには編集可能な属性が含まれ、入力のタイプや使用する入力文字列のヘッダーに関する情報も含まれます。たとえば、次のようになります。

```
<String name="uid" value="string|User Name:"/>
<String name="password" value="password|User Password:"/>
```

コレクション内の名前は、対応するディスプレイプロファイル SSO アダプタ属性の名前と一致する必要があります。項目の値の部分には、「|」という文字で区切られた 2 つの情報が含まれています。値文字列の最初の部分は、その属性の表示タイプを指定します。属性の値文字列の 2 番目の部分は、その項目の隣に表示される文字列を指定します。次に、表示タイプが HTML の GUI 項目にどのように対応するかを示します。

- **string:** 英数字を入力できるテキストフィールドを作成します。
- **password:** 入力内容が「*」に置き換えて表示されるパスワードフィールドを作成します。
- **check:** チェックボックスを作成します。
- **select:** 選択ボックスを作成します。すべての選択項目は、値と表示テキストのリストが設定された、対応するコレクションを必要とします。

select 表示タイプのすべての項目は、返される値とオプションの表示値 (文字列) のリストが設定された、対応するコレクションを必要とします。このコレクションの名前は、その属性の名前と、SelectOptions というテキストから構成されます。たとえば、MailProvider の sortOrder 属性であれば、コレクション名は sortOrderSelectOptions となります。

```
<Collection name="sortOrderSelectOptions" advanced="false"
merge="replace" lock="false" propagate="true">
```

```
<String name="top" value="Most recent at top"/>
<String name="bottom" value="Most recent at bottom"/>
</Collection>
```

編集ページの HTML テンプレート

通信チャネルプロバイダの編集ページの作成には、9つの HTML テンプレートが使用されます。これらのテンプレートは、特定のブラウザの GUI タイプに対応できるように、汎用性が高く作成されています。ほとんどは、編集ページ内の特定の HTML 入力に関連するものです。edit-start.template と edit-end.template は例外です。これらのテンプレートには、ページのレイアウトに使用されるほとんどの HTML コードが含まれます。393 ページの表 17-2 は、各テンプレートの名前と、それが GUI タイプとどのように関連するかを示しています。一部のテンプレートは、属性の開始、終了、分割に使用されます。各通信チャネル用のこれらのテンプレートは、次の場所に格納されています。

```
/etc/opt/SUNWps/desktop/default/ChannelName_Provider/html
```

たとえば、カレンダーチャネルの編集ページのテンプレートは、次の場所にあります。

```
/etc/opt/SUNWps/desktop/default/CalendarProvider/html
```

表 17-2 通信チャネル編集ページのテンプレート

テンプレート	説明
edit-start.template	編集ページの HTML テーブルの開始を指定します。
edit-checkbox.template	チェックボックス項目用の汎用テンプレートを提供します。
edit-separate.template	ディスプレイプロファイル属性と SSO 属性を分割します。
edit-end.template	編集ページの HTML テーブルの終了を指定します。
edit-password.template	パスワード項目用の汎用テンプレートを提供します。
edit-string.template	テキスト項目用の汎用テンプレートを提供します。
edit-select.template	選択項目用の汎用テンプレートを提供します。
edit-selectoption.template	選択オプション用の汎用テンプレートを提供します。この方法では、オプションをディスプレイプロファイルから動的に生成できます。
edit-link.template	ユーザーがクライアントの表示属性を編集するためのリンク生成用のテンプレートを提供します。

ディスプレイプロファイルの例

この例は、エンドユーザーが通信チャネルの編集ページで特定の機能のエントリを変更し、各自のポータルデスクトップで通信チャネルがどのように設定、表示されるかを変更できるようにするために、SSO アダプタの特定の属性がそれぞれに対応するディスプレイプロファイルの属性とどのように連携して機能するかを示しています。

394 ページのコード例 17-2 に示す SSO アダプタテンプレートは、メールチャネルの例です。SSO アダプタテンプレートには、マージされた属性が 2 つ含まれます。

- uid: ユーザー ID
- password: ユーザーパスワード

マージされた属性は、エンドユーザーが指定できる属性です。管理者は、どの属性をマージされた属性にするか、つまり、エンドユーザーがどの属性を編集できるようにするかを決定します。

コード例 17-2 SSO アダプタテンプレートの例

```
default|imap:///&configName=MAIL-SERVER-TEMPLATE
&encoded=password
&default=protocol
&default=clientProtocol
&default=type
&default=subType
&default=ssoClassName
&default=smtpServer
&default=clientPort
&default=host
&default=port
&merge=username
&merge=userpassword
&clientProtocol=http
&type=MAIL-TYPE
&subType=sun-one
&ssoClassName=com.sun.ssoadapter.impl.JavaMailSSOAdapter
&smtpServer=example.sun.com
&clientPort=80
&host=company22.example.com
&port=143
```

395 ページのコード例 17-3 は、チャネルの `ssoEditAttributes` のディスプレイプロファイル XML コード例を示しています。

SSO アダプタテンプレートで管理者が属性に `merge` を設定した後に、ディスプレイプロファイル内の属性を編集し、その属性がエンドユーザーの編集ページにどのように表示され、エンドユーザーがそれをどのように編集できるかを再設定することができます。管理者は、ディスプレイプロファイルの適切なコレクションを編集し、エンド

ユーザーが必要な情報を照会する方法を決定できます。たとえばこの例では、管理者は User Name を What is your user name? という質問に置き換えることができます。通常は、「|」記号の前に指定する属性表示タイプとして、string を使用します。ただし、管理者が password タイプなど、別のタイプを指定することも可能です。

コード例 17-3 メールチャネルのディスプレイプロファイル XML コードの例

```
<Channel name="SampleMailChannel" provider="MailProvider">
  <Properties>
    <Collection name="ssoEditAttributes">
      <String name="username" value="string|User Name:"/>
      <String name="userpassword" value="password|User Password:"/>
    </Collection>
  </Properties>
</Channel>
```

このメールチャネル編集ページの例では、エンドユーザーに次のテキストフィールドが表示されます。

- User Name:
- User Password:

通信チャネルのタイプとしてエンドユーザーが複数インスタンスを設定できるようにする

エンドユーザーまたは管理者は、複数のタイプの通信チャネルを作成できます。エンドユーザーが複数のタイプの通信チャネルを作成するには、「コンテンツ」ページの「新しいチャネルの作成」リンクを使用する必要があります。

管理者は、組織、ロール、またはグループ用の複数のチャネルを作成できます。管理者が、アドレス帳コンポーネントの第2インスタンスのように特定のコンポーネントの複数のインスタンスを使用できるようにすると、管理者は、エンドユーザーが各自のポータルデスクトップで第2アドレス帳チャネルを設定できるようにすることができます。

管理者は、新しい通信チャネルのタイプごとに SSO アダプタテンプレートを作成するか、1つの SSO アダプタテンプレートを使用して、チャネルごとに複数の SSO アダプタ設定を作成することができます。SSO アダプタの詳細については、[453 ページの付録 A 「SSO アダプタテンプレートと設定」](#) を参照してください。

管理者が完了した設定の量によっては、エンドユーザーは多数の設定を行う必要がありません。管理者は、アプリケーション設定編集機能を使用して、これらの設定を行うことができます (391 ページの「[アプリケーション設定の編集: 通信チャネル編集ページの設定](#)」参照)。

2つのアドレス帳チャネルを作成するには、それぞれが異なる SSO アダプタテンプレート参照するようにします。これで、両方のアドレス帳チャネルを元のページに追加できます。同様に、1つの SSO アダプタテンプレートと2つの SSO アダプタ設定 (ダイナミック) を作成することもできます。SSO アダプタテンプレートはサーバーの設定をユーザーが定義できる値 (merge) として定義し、SSO アダプタ設定がこれらのサーバー設定を指定します。

エンドユーザーが必要に応じてサーバーを設定できるように、異なるサーバーのアドレス帳を設定するには、次の手順を実行します。

1. SSO アダプタテンプレートで、ユーザーが定義できる値 (merge) としてサーバー情報を指定します。詳細については、453 ページの付録 A 「[SSO アダプタテンプレートと設定](#)」を参照してください。
2. チャネルのディスプレイプロファイルの `ssoEditAttributes` コレクションで、どの属性を編集可能にするかを指定します。詳細については 391 ページの「[アプリケーション設定の編集: 通信チャネル編集ページの設定](#)」を参照してください。ディスプレイプロファイルに関する情報については、『[Sun Java System Portal Server 6 2005Q1 Desktop Customization Guide](#)』を参照してください。

管理者のプロキシ認証：エンドユーザーによる信用情報の設定の禁止

アドレス帳、カレンダー、メールの各チャネルで、管理者のプロキシ認証を有効にすることができます。Sun Java System Portal Server と Sun Java System メッセージサービス (Messaging Server および Calendar Server) の間でプロキシ認証のサポートを拡張することで、エンドユーザーはチャネルの編集ページにアクセスせずに、各自の信用情報 (ユーザー名とユーザーパスワード) を入力できるようになります。エンドユーザーの信用情報の代わりに、管理者の信用情報が使用され、SSO アダプタテンプレートに格納されます。テンプレート内では、管理者のユーザー ID は `proxyAdminUid` 属性の値として格納され、管理者のパスワードは `proxyAdminPassword` 属性の値として格納されます。ユーザーがチャネルを起動するたびに、これらの値を使用してチャネルと、それに対応するバックエンドサーバーが接続されます。このとき、ユーザーの命名属性もバックエンドサーバーに送信されます。管理者プロキシ認証の命名属性の使用については、398 ページの表 17-3 の「`userAttribute` プロパティ」を参照してください。

Sun Java System Instant Messaging Server、Microsoft Exchange Server、IBM Lotus Notes サーバーには、プロキシ認証を設定できません。

注意：複数のエンドユーザーが1つのメールアカウントにダイレクトされる可能性
Access Manager と Portal Server では、管理者は複数のユーザーが組織内で同じユーザー ID を持つように設定することができます。たとえば、組織に2つのサブ組織があり、それぞれが enduser22 という名前のエンドユーザーを持つことができます。Sun Java System の通信チャネルで管理者のプロキシ認証を有効にし、エンドユーザーの命名属性をデフォルト (uid) に設定すると、両方のユーザーが同じバックエンドユーザーアカウントにアクセスする可能性が生じます。管理者のプロキシ認証では、ユーザーの命名属性を SSO アダプタテンプレートで変更することができます。たとえば、この属性を従業員間で一意の属性 (従業員番号など) に変更することで、ポータルのエンドユーザーが正しいバックエンドサーバーアカウントに確実にアクセスできるようになります。

プロキシ認証の設定方法の概要

アドレス帳、カレンダー、メールチャネルの管理者プロキシ認証を有効にするには、Sun Java System Access Manager の管理コンソールから SSO アダプタテンプレートにアクセスし、各種 Sun Java System 通信サーバーにアクセスする必要があります。具体的には、次の操作が必要です。

- SSO アダプタテンプレートの編集。
 - SSO アダプタテンプレートで、アドレス帳、カレンダー、メールの各チャネルに適用される文字列を編集する必要があります。この文字列で違いが際立っている項目の一つに、使用プロトコルがあります。
 - アドレス帳チャネルは LDAP プロトコルを使用します。
 - カレンダーチャネルは HTTP プロトコルを使用します。
 - メールチャネルは IMAP または POP プロトコルを使用します。
- Sun Java System Messaging Server にアクセスし、アドレス帳チャネルとメールチャネルのプロキシ認証を有効にします。
- Sun Java System Calendar Server にアクセスし、カレンダーチャネルのプロキシ認証を有効にします。

プロキシ認証と SSO (シングルサインオン) アダプタテンプレート

管理者プロキシ認証に合わせて SSO アダプタテンプレートを編集する

1. インターネットブラウザで `http://hostname:port/amconsole` (たとえば `http://psserver.company22.example.com:80/amconsole`) と入力し、Sun Java System Access Manager の管理コンソールにログインします。
2. 「サービス設定」タブをクリックします。設定できるサービスのリストがナビゲーションパネルに表示されます。

3. 「SSO アダプタの設定」の下の「SSO アダプタ」を選択します。SSO アダプタを編集するためのページがデータパネルに表示されます。
4. 管理者のプロキシ認証を有効にするチャネルの文字列をクリックします。
5. 設定を説明するフィールドの内側をクリックします。
6. 管理者のプロキシ認証に必要な情報でキーを削除します。

表 17-3 は、管理者のプロキシ認証をサポートするために SSO アダプタテンプレートで編集が必要なプロパティを示しています。

表 17-3 管理者プロキシ認証に関係する SSO アダプタテンプレートのプロパティ

プロパティ	値	説明
enableProxyAuth	true false	この属性に関連付けられている値はフラグで、プロキシ認証が有効であるかどうかを示します。true の場合は、SSO アダプタとアプリケーションアダプタはプロキシ認証を行います。たとえば次のように指定します。 &enableProxyAuth=true
proxyAdminUid	(設定可能)	この属性に関連付けられている値は、管理者のユーザー名です。たとえば次のように指定します。 &proxyAdminUid=ServiceAdmin
proxyAdminPassword	(設定可能)	この属性に関連付けられている値は、管理者のユーザーパスワードです。たとえば次のように指定します。 &proxyAdminPassword=mailpwd
userAttribute	(設定可能)	この属性に関連付けられている値は、管理者の命名属性です。この値は、ユーザーのレコード (ディレクトリ内のユーザーエントリ) に設定されている属性にマッピングされます。一般に、このレコードにはユーザー ID (uid) や従業員番号など、いくつかの属性が設定されています。デフォルトでは、命名属性は uid に設定されます。たとえば次のように指定します。 &userAttribute=uid SSO アダプタテンプレートを編集することで、命名属性に従業員番号などの別の属性にマッピングすることができます。

次に示すように、前述の 4 つのプロパティは、SSO アダプタテンプレートの文字列にもう一度表示されます。プロパティの設定は、default または merge に設定できます。次の例では、すべてが default に設定されています。

プロパティ	値	例
enableProxyAuth	default	&default=enableProxyAuth

表 17-3 管理者プロキシ認証に関する SSO アダプタテンプレートのプロパティ (続き)

プロパティ	値	説明
proxyAdminUid	default	&default=proxyAdminUid
proxyAdminPassword	default	&default=proxyAdminPassword
userAttribute	default	&default=userAttribute

プロキシ認証と通信サーバー

管理者のプロキシ認証に合わせて Sun Java System Messaging Server を設定する

1. Sun Java System Messaging Server ソフトウェアのホストにログインし、スーパーユーザーになります。
2. 次のコードを入力します。

```
MessagingServer-base/msg-instance-name/configutil -o
service.http.allowadminproxy -v yes
```

3. Messaging Server を再起動します。

configutil の実行方法とサーバーの再起動方法については、『Sun Java System Messaging Server 管理者ガイド』を参照してください。

管理者のプロキシ認証に合わせて Calendar Server を設定する

1. Sun Java System Calendar Server ソフトウェアのホストにログインし、スーパーユーザーになります。
2. 任意のエディタで次のファイルを開きます。

```
CalendarServer-base/cal/bin/config/ics.conf
```

3. 次に示す属性をこのとおりに設定します。

```
service.http.allowadminproxy = "yes"
```

4. カレンダーサーバーを再起動します。

詳細な実行方法とサーバーの再起動方法については、『Calendar Server 管理者ガイド』を参照してください。

認証不要ポータルデスクトップ用に読み取り専用通信チャネルを設定する

認証不要 (認証なしの匿名) ポータルデスクトップは、読み取り専用の通信チャネルをサポートします。

読み取り専用通信チャネルに関する情報と注意点

認証なしの匿名ポータルデスクトップ用に、アドレス帳、カレンダー、メールチャネルへの読み取り専用アクセスを設定できます。エンドユーザーは、ポータルデスクトップにアクセスするだけで、つまりインターネットブラウザに次の URL を入力するだけで、読み取り専用通信チャネル内の情報にアクセスできます。

`http://hostname.domain:port/portal/dt`、たとえば、
`http://pssserver.company22.example.com:80/portal/dt`

エンドユーザーは、管理者が設定した読み取り専用通信チャネルにログインなしでアクセスすることができます。ただし、通常はエンドユーザーによるこれらのチャネルの編集は禁じられます。匿名ログインの有効化など、認証なしポータルデスクトップの詳細については、『Sun Java System Portal Server 6 2005Q1 Desktop Customization Guide』を参照してください。

カレンダーチャネルは、複数のユーザーによって共有されることが最も多いチャネルです。次に、全員が共有できるように読み取り専用カレンダーチャネルを設定する手順について説明します。この例では、共有しているカレンダーは、ユーザー *library* に属しています。公開されている読み取り専用カレンダーのタイトルは *Library Schedule* です。次のカレンダーの設定は、方法の 1 つにすぎません。Sun Java System Calendar Server 用にユーザーを設定する方法の詳細については、『Sun Java System Calendar Server 管理者ガイド』の「*csuser* コマンド」の説明で「*create userid* オプション」を参照してください。

カレンダーユーザーを設定する

1. 次のようなコマンドを発行してカレンダーユーザーを作成します。

```
csuser -g Library -s Admin -y libadmin -l en -m  
libadmin@library.com -c librarySchedule create libadmin
```

ここで、ユーザー *libadmin* は、名が *Library*、姓が *Admin*、パスワードが *libadmin*、設定言語が *en* (英語)、電子メールアドレスが *libadmin@library.com*、カレンダー ID が *librarySchedule* です。

2. 次に対してアクセス権を全ユーザーが読み取り可能になるように設定します。

```
libadmin:librarySchedule
```

アクセス権の設定は、管理者が *cscal* ユーティリティを使用するか、エンドユーザーが *Calendar Express* を使用して行います。

読み取り専用通信チャネルを設定する

1. エンドユーザー (この場合は認証なしの匿名ユーザー) の設定を編集し、カレンダー SSO アダプタ設定を作成します。
 - a. インターネットブラウザで `http://hostname:port/amconsole` (たとえば `http://psserver.company22.example.com:80/amconsole`) と入力し、Sun Java System Access Manager の管理コンソールにログインします。
 - b. 「アイデンティティ管理」タブをクリックし、ナビゲーションパネルに「表示」ドロップダウンリストを表示します。
 - c. 「表示」ドロップダウンリストの「ユーザー」をクリックします。
 - d. 必要に応じて認証なしの匿名ユーザーまでスクロールし、隣にある矢印をクリックします。認証なしの匿名ユーザーのページがデータパネルに表示されます。

これで SSO アダプタサービスを認証なしの匿名ユーザーに追加できるようになります。
 - e. 認証なしの匿名ユーザーのページの「表示」ドロップダウンリストで「サービス」をクリックし、利用可能なサービスを表示します。
 - f. 「追加」をクリックします。
 - g. SSO アダプタのチェックボックスをクリックします。
 - h. 「保存」をクリックします。
2. 認証なしの匿名ユーザー用にカレンダー SSO アダプタ設定を作成します。
 - a. Sun Java System Access Manager 管理コンソールにログインしていない場合は、ログインします。
 - b. 「アイデンティティ管理」タブをクリックし、ナビゲーションパネルに「表示」ドロップダウンリストを表示します。
 - c. 「表示」ドロップダウンリストから「サービス」を選択し、設定可能なサービスのリストを表示します。
 - d. SSO アダプタの横の矢印をクリックして、データパネルに「SSO アダプタ」ページを開きます。
 - e. 設定を説明する空白のフィールドに、グループに関連する SSO アダプタ設定文字列 (ユーザー ID とパスワード) を入力します。参照のために、一般的な設定を次に示します。文字列に指定できる属性は、Sun Java System Portal Server SSO アダプタテンプレートをもとに設定したかによって異なります。デフォルトでは、SSO アダプタテンプレートでは、ユーザーが次の情報を指定していることを前提としています。
 - host
 - port

- client port
- uid
- password

ページを表示したときに設定を説明するフィールドが空でなかった場合は、フィールド内のテキストをすべて選択し、削除してから、次の形式で文字列を入力します。

```
default|undef://?uid:password@host:port/?  
configName=configuration-name  
&configDesc=configuration-description
```

たとえば、次のようになります。

```
default|undef://?libadmin:libadmin@example.com:3080/?  
configName=sunOneCalendar_librarySchedule  
&configDesc=SUN-ONE-CALENDAR
```

- f. 「追加」をクリックします。
 - g. 「保存」をクリックします。
3. 新しく作成した SSO アダプタ設定に基づいて、認証なしの匿名ユーザーの新規カレンダーチャネルを作成します。
- a. Sun Java System Access Manager 管理コンソールにログインしていない場合は、ログインします。
 - b. 「アイデンティティ管理」タブをクリックし、ナビゲーションパネルに「表示」ドロップダウンリストを表示します。
 - c. 「表示」ドロップダウンリストの「ユーザー」をクリックします。
 - d. 必要に応じて認証なしの匿名ユーザーまでスクロールし、隣にある矢印をクリックします。認証なしの匿名ユーザーのページがデータパネルに表示されます。
- これで認証なしの匿名ユーザー用に新規カレンダーチャネルを作成できます。
- e. 認証なしの匿名ユーザーのページの「表示」ドロップダウンリストで「ポータルデスクトップ」をクリックし、「編集」リンクを表示します。
 - f. 「編集」リンクをクリックします。
 - g. 「チャネルおよびコンテナを管理」リンクをクリックします。
 - h. 「チャネル」セクションまでスクロールし、「新規」をクリックします。
 - i. 「チャネル名」フィールドに名前を入力します。たとえば、次のようになります。

```
LibraryScheduleChannel
```

- j. プロバイダのドロップダウンリストから、適切なプロバイダを選択します。この例では「Calendar Provider」です。
- k. 「了解」をクリックします。「チャネルおよびコンテナを管理」ページに戻ります。

これでチャネルのプロパティを編集できます。
- l. 「チャネル」セクションまでスクロールし、新しく作成したチャネルの隣にある「プロパティの編集」をクリックします。たとえば、次のようになります。
`LibraryScheduleChannel`
- m. フィールドを適切に編集します。たとえば、次のようになります。
 - o title: Library Schedule
 - o description: Library Schedule
 - o ssoAdapter: sunOneCalendar_librarySchedule
 - o loadSubscribedCalendars: false (チェックマークなし)
 - o is editable: false (チェックマークなし)
- n. 必要に応じてページをスクロールし、「保存」をクリックします。

これで認証なしの匿名ユーザーのポータルデスクトップに新規カレンダーチャネルを追加できます。
- o. ページ上部近くにある「トップ」をクリックします。「チャネルおよびコンテナを管理」ページに戻ります。
- p. 「コンテナチャネル」セクションまで下にスクロールし、新しいチャネルを追加するコンテナのリンクをクリックします。たとえば、`MyFrontPageTabPanelContainer` をクリックします。対応する「プロパティの編集」リンクをクリックしないように注意してください。
- q. 「チャネル管理」の見出しの下に表示される、作成したチャネルの名前をクリックします。たとえば、「使用可能」リストに表示される「`LibraryScheduleChannel`」をクリックします。
- r. 「コンテンツページでエンドユーザーが使用可能」リストまたは「ポータルデスクトップで表示可能」リストにチャネルを追加します。

チャネルを追加するリストの上にある「追加」ボタンをクリックします。
- s. ページを上スクロールし、「チャネル管理」という見出しの下の「保存」をクリックします。
- t. Web コンテナを再起動します。

Microsoft Exchange Server または IBM Lotus Notes の設定

Sun Java System Portal Server 6 は、通信チャネル用に Sun Java System Messaging Server と Sun Java System Calendar Server をサポートするほか、Microsoft Exchange Server と IBM Lotus Notes サーバーもサポートします。

エンドユーザーが Microsoft Outlook Web Access ソリューションにアクセスできるよう、Sun Java System Portal Server と連携するように Microsoft Exchange Server を設定できます。エンドユーザーがポータルデスクトップで各チャネルの「アドレス帳を起動」、「カレンダーを起動」、「メールを起動」をクリックすると、このソリューションにアクセスできるようになります。

同様に、Sun Java System Portal Server と連携するように IBM Lotus Notes サーバーを設定することで、エンドユーザーはアドレス帳、カレンダー、メールチャネルから IBM Lotus Domino Webmail ソリューションにアクセスできるようになります。

注 Microsoft Exchange Server と IBM Lotus Notes サーバーは、管理者プロキシ認証とシングルサインオンをサポートしていません。このシングルサインオンの制約により、これらのサーバーの 1 つの接続されたチャネルをエンドユーザーが起動する場合、接続が確立する前にエンドユーザーは各自の信用情報を再入力する必要があります。

アドレス帳、カレンダー、メール用に Microsoft Exchange 5.5 Server を設定する

1. ドメインの管理者として PDC (Primary Domain Controller) にログインします。
2. 「Start」、「Programs」、「Administrative Tools」、「User Manager for Domains」の順に選択し、MAXHost というユーザー名でアカウントを作成します。
3. 「Groups」を選択し、「Administrators」グループと「Domain Admins」グループに MAXHost を追加します。
4. MAXHost が MAIL_HOST、ドメインコントローラ、MAX_HOST にローカルにログインできることを確認します。
5. パスワードを設定します。
6. MAXHost として Exchange 5.5 (MAIL_HOST) にログインします。
7. 「Start」、「Programs」、「Microsoft Exchange」、「Microsoft Exchange Administrator」の順に選択します。
8. 各エンドユーザーにメールボックスへのアクセス権を設定します。

9. アクセス権のタブを表示するには、「Tools」、「Options」、「Permissions」の順に選択し、「All Objects」の「Show Permissions Page」を有効にします。
10. ユーザー名をダブルクリックします。
11. アクセス権のタブを選択し、アクセス権のページの「追加」をクリックして MAXHost を追加し、ロールは「User」のまま残します。
通信チャネルにアクセスするユーザーごとに手順 9～11 を繰り返します。
12. 次のディレクトリにある ocxhost.zip ファイルを解凍します。
PortalServer-base/SUNWps/export
ファイルを解凍すると、次の形式のファイルが生成されます。

```
Archive: ocxhost.zip
creating: ocxhost
creating: ocxhost/international
inflating: ocxhost/international/ocxhostEnglishResourceDll.dll
inflating: ocxhost/ocxhost.exe
```

13. 次の手順を実行し、ocxhost を登録します。
 - a. ocxhost.exe の場所を特定します。
 - b. 「Start and Run」を選択します。
 - c. 「Run」ウィンドウに次のように入力します。
ocxhost.exe /multipleuse
14. ocxhost ユーティリティのプロパティを設定するには、次の手順に従います。
 - a. dcomcnfg ユーティリティを使用して、ocxhost ユーティリティの適切な DCOM 設定を指定します。次のように指定します。
 - I. 「Start and Run」を選択します。
 - II. **dcomcnfg** と入力し、「了解」をクリックします。
 - b. 「Distributed COM Configuration Properties」ダイアログボックスで、次のように操作します。
 - I. 「Default Properties」タブを選択します。
 - 「Enable Distributed COM on the computer」チェックボックスにチェックマークを付けます。
 - デフォルトの「Authentication Level」を「Connect」に設定します。
 - デフォルトの「Impersonation Level」を「Identify」に設定します。

- II. 「Applications」 タブを選択します。
 - III. 「Properties」 ダイアログで `ocxhost` ユーティリティをダブルクリックします。
「ocxhost properties」 ウィンドウが表示されます。
 - IV. 「Location」 タブの下にある 「Run Application on this Computer」 にチェックマークを付けます。
 - V. 「Security」 タブの下にある 「Use custom access permissions」、「Use custom launch permissions」、「Use custom configuration permissions」を設定します。
 - VI. 「Access」、「Launch」、「Configuration」 の設定で 「Edit」 を選択し、アクセス制御リスト (ACL) に次のユーザーが含まれていることを確認します。
 - Interactive
 - Everyone
 - System
 - VII. 「ocxhost properties」 ウィンドウの 「Identity」 タブの下にある 「User」 を選択します。
 - VIII. 「Browse」 を選択して、MAXHost を探します。
 - IX. パスワードとパスワードの確認入力を入力します。
- c. 「了解」 を選択します。
- ocxhost DCOM コンポーネントの設定が完了し、Exchange Server と通信する準備が整います。

アドレス帳、カレンダー、メール用に Microsoft Exchange 2000 Server を設定する

Portal Server を、複雑な Microsoft Windows 2000 ドメイン設定の Exchange Server 2000 環境から、カレンダーデータにアクセスするように設定する場合、`ocxhost.exe` を、MAX_HOST と呼ばれる専用のシステムにインストールする必要があります。

この方法には 2 つの利点があります。

- ユーザーがポータルからカレンダーにアクセスできない場合、障害追跡を容易にします。
- ファイアウォールがポータルサーバーと Microsoft Windows ドメイン間に存在する場合、より制限的なセキュリティ設定が可能になります。

Exchange Server がクラスタである、またはフロントエンドおよびバックエンド Exchange Server が設定されている、あるいはその両方の場合、「複雑な」ドメインになります。また、Microsoft Windows ユーザーと、同じエンドユーザーの Exchange Mailbox が異なるドメインの場合も、「複雑な」ドメインになります。

次の手順では、以下のように仮定します。

MAX_HOST: Outlook 2000 を実行する専用の Microsoft Windows 2000 システムの名前で、ocxhost.exe がインストールされています。

MAIL_HOST: エンドユーザーのメールボックスが存在する Exchange Server です。

PORTAL: Java Enterprise System Portal Server 2005Q1 です。

DOMAIN: MAX_HOST および MAIL_HOST がある Microsoft Windows ドメインです。

専用の Microsoft Windows 2000 System (MAX_HOST) を設定する場合、次の要件と仮定を確認してください。

- Microsoft Windows 2000 Server SP3 または Microsoft Windows 2000 Professional。
 - CDO が有効な Microsoft Outlook 2000。
 - オペレーティングシステムおよび Outlook 2000 がインストールされている。IP アドレスが割り当てられ、Exchange Server と同じドメインに新しいホストがある。
1. ドメインに User MAXhost を作成します。
 - a. ドメインの管理者として、ホスト (MAX_HOST) にログインします。
 - b. 「Start」、「Programs」、「Administrative Tools」、「Active Directory Users」、「Computers」の順に選択し、MAXHost というユーザー名でドメインアカウントを作成します。
 - c. 「User」、「Properties」、「Member of」の順に選択して、グループ管理者 (ローカル) を追加します。
 - d. MAXHost が MAIL_HOST および MAX_HOST にローカルにログインできることを確認します。
 - e. パスワードを設定します。
 2. MAXHost ユーザー用に Outlook を設定します。
 - a. ドメインユーザー MAXHost として MAX_HOST システムにログインします。
 - b. Outlook を起動して、ユーザー MAXHost 用の Outlook プロファイルを設定します (必要に応じて、Microsoft マニュアルを参照)。
 - c. MAXHost ユーザー用に Outlook の設定が完了したら、Outlook を閉じます。
- 注 : Outlook は、ocxhost.exe と同時には動作できません。

3. アドレス帳、カレンダー、およびメール用に Microsoft Exchange Server を設定します。
 - a. MAXHost として Exchange 2000 Server (MAIL_HOST) にログインします。
Exchange 2000 Front-End Server を使用する場合、MAXHost としてフロントエンドサーバーにログインします。
 - b. 「Start」、「Programs」、「Microsoft Exchange」、「Active Directory Users」、「Computers」の順に選択します。
 - c. 各エンドユーザーにメールボックスへのアクセス権を設定します。
 - d. 「View」、「Advance Features」の順に選択します。
 - e. ユーザー名をダブルクリックします。
 - f. 「Exchange Advanced」タブを選択し、「Mailbox Rights」を選択します。
 - g. MAXHost を追加し、MAXHost 完全アクセスを提供します。
通信チャネルにアクセスするユーザーごとに、手順 d ~ 手順 g を繰り返します。
4. MAX_HOST に、ocxhost.exe をインストールします。
 - a. ドメインユーザー MAXHost として、MAX_HOST にログインします。
 - b. 次のディレクトリにある ocxhost.zip ファイルを解凍します。
PortalServer-base/SUNWps/export
ファイルを解凍すると、次の形式のファイルが生成されます。
 - o Archive: ocxhost.zip
 - o creating: ocxhost
 - o creating: ocxhost/international
 - o inflating: ocxhost/international/ocxhostEnglishResourceDll.dll
 - o inflating: ocxhost/ocxhost.exe
 - c. 次の手順を実行し、ocxhost を登録します。
 - I. ocxhost.exe ファイルの場所を特定します。
 - II. 「Start and Run」を選択します。
 - III. 「ocxhost.exe /multipleuse」と入力して、「了解」を選択します。
注: このコマンドが実行されるたびに、次の手順で説明されている DCOM 設定が消去されて、再設定が必要になるため、この登録は 1 回のみ行います。

5. dcomcnfg ユーティリティを使用して、ocxhost ユーティリティに必要な DCOM 設定を指定します。
 - a. 「Start and Run」を選択します。
 - b. 「dcomcnfg」と入力し、「了解」をクリックします。
 - c. 「Distributed COM Configuration Properties」ダイアログボックスで、「Default Properties」タブを選択し、次の設定を行います。
 - 「Enable Distributed COM on the computer」チェックボックスにチェックマークを付けます。
 - デフォルトの「Authentication Level」を「Connect」に設定します。
 - デフォルトの「Impersonation Level」を「Identify」に設定します。
 - d. 「Applications」タブを選択します。
 - e. 「Properties」ダイアログで ocxhost ユーティリティをダブルクリックします。「ocxhost properties」ウィンドウが表示されます。
 - f. 「Location」タブの下にある「Run Application on this Computer」にチェックマークを付けます。
 - g. 「Security」タブの下にある「Use custom access permissions」、「Use custom launch permissions」、および「Use custom configuration permissions」を設定します。
 - h. 「Access」、「Launch」、「Configuration」の設定で「Edit」を選択し、アクセス制御リスト (ACL) に次のユーザーが含まれていることを確認します。
 - Interactive
 - Everyone
 - System
 - i. 「ocxhost properties」ウィンドウの「Identity」タブの下にある「User」を選択します。
 - j. 「Browse」を選択して、MAXHost を探します。
 - k. パスワードとパスワードの確認入力を入力します。
 - l. 「了解」を選択します。

ocxhost DCOM コンポーネントの設定が完了し、Exchange Server と通信する準備が整います。Portal Server からの最初のアクセスが発生するときに、ocxhost DCOM コンポーネントは、RPC コールによって起動されます。
6. MAXHost ユーザーグループを変更します。

セキュリティ上の理由から、管理グループからドメインユーザーを削除する場合は、次のようにします。

- a. ログアウトしてから、管理者として再び MAX_HOST にログインします。
- b. ローカルの管理グループからユーザー MAXHost を削除します (それをドメインユーザーグループに割り当てる)。

注 ファイウォールは Portal と MAX_HOST の間で、使用しないでください。
(ダイナミックポートを使用する RPC コールは、Portal Server から ocxhost.exe への通信用に使用される)。

ファイウォールは、MAX_HOST と MAIL_HOST の間で使用しないでください。

カレンダー用の SSO アダプタ

ocxhost.exe (MAX_HOST) 専用のサーバーを使用している場合、カレンダー用に SSO アダプタを設定します。

1. SSO アダプタテンプレートを作成します。
 - a. Access Manager 管理コンソールにログインします。
 - b. 「サービス設定」タブを選択します。
 - c. 「SSO アダプタ」を選択します。
 - d. 「新規」を選択します。
 - e. 新しいテンプレート用の名前を入力し、リストから既存の EXCHANGE-CALENDAR を選択します。
 - f. 「次へ」を選択します。
 - g. ocxHost 用の行に、ocxhost.exe が存在するシステム (この場合は MAX_HOST) の dns-name または IP-Address を入力します。
 - h. 「保存」をクリックします。
2. 組織用に SSO アダプタ設定を作成します。
 - a. 「アイデンティティ管理」タブから、組織を選択します。
 - b. スクロールダウンメニューから「サービス」を選択します。
 - c. 「SSO アダプタ」を選択します。
 - d. 「SSO アダプタ設定」から「新規」を選択します。
 - e. 設定用の名前を入力し、以前作成したテンプレートを選択します。
 - f. 「次へ」を選択します。

- g. 必要に応じてプロパティを修正します。
MAIL_HOST (DNS 名または IP-Address) であるデフォルトのホスト名を入力するか、空白のままにします。
- h. 「保存」を選択し、「変更を保存しました」のメッセージを確認します。
SSO アダプタテンプレートと設定の使用に関する説明については、[付録 A 「SSO アダプタテンプレートと設定」](#) も参照してください。

ocxhost.exe をアンインストールする

次の手順を実行し、ocxhost を登録解除します。

1. ocxhost.exe の場所を特定します。
2. 「Start and Run」を選択します。
3. 「Run」ウィンドウに次のように入力します。
`ocxhost.exe /unregserver`
4. ocxhost.exe ファイルと ocxhostEnglishResourceDll.dll ファイルを削除します。

アドレス帳、カレンダー、メール用に Lotus Domino Server を設定する

1. 「Start」、「Programs」、「Lotus Applications」、「Lotus Administrator」の順に選択して、Lotus Administrator を開きます。
2. 「Administration」、「Configuration」、「Server」、「Current Server Documents」の順に選択します。
3. 「Security」タブで、次のように設定します。
 - a. 「Java/COM Restrictions」の下で、「Run restricted Java/Javascript/COM」と「Run unrestricted Java/Javascript/COM」を * に設定します。
 - b. 「Security Settings」の下で、次のように設定します。
 - 「Compare Notes Public keys against those stored in Directory」を「No」に設定します。
 - 「Allow anonymous Notes connections」を「No」に設定します。
 - 「Check Passwords on Notes IDs」を「Disabled」に設定します。
 - c. 「Server Access」の下で、「Only allow server access to users listed in this Directory」を「No」に設定します。
 - d. 「Web Server Access」の下で、「Web Server Authentication」をセキュリティの低い「More Name Variations」に設定します。
4. 「Ports」タブで、次のように設定します。

- a. 「Notes Network Ports」タブを選択し、TCPIP が「ENABLED」であることを確認します。
- b. 「Internet Ports」タブと「Web」タブを選択します。
 - I. TCP/IP ポートの状態が「Enabled」であることを確認します。
 - II. 「Authentication」オプションの下で、「Name and password」と「Anonymous」が「Yes」に設定されていることを確認します。
- c. 「Directory」タブを選択し、次の設定を確認します。
 - TCP/IP ポートの状態が「Enabled」である。
 - 「Authentication」オプション項目の「Name and password」と「Anonymous」が「Yes」に設定されている。
 - SSL ポートの状態が「Disabled」に設定されている。
- d. 「Mail」タブを選択し、次の設定を確認します。
 - TCP/IP ポートの状態が「Enabled」である。
 - 「Authentication」オプションの「Name and Password」と「Anonymous」が次のように設定されている。

	メール (IMAP)	メール (POP)	メール (SMTP 受信)	SMTP (送信)
Name and Password	あり	あり	なし	なし
Anonymous	なし	なし	あり	なし

- e. 「IIOp」タブを選択し、次の設定を確認します。
 - TCP/IP ポートの状態が「Enabled」である。
 - 「Authentication」オプション項目の「Name and password」と「Anonymous」が「Yes」に設定されている。
 - TCP/IP ポート番号が 0 に設定されていない。これは、63148 に設定する必要があります。
 - SSL ポートの状態が「Disabled」に設定されている。
5. 「Internet Protocols」タブを選択し、「IIOp」サブタブを選択します。「Number of threads」が 10 以上に設定されていることを確認します。
6. 設定を保存して閉じます。
7. Domino Server のコンソールで次のように入力し、サーバーを再起動します。

restart server

サーバーを再起動すると、設定が有効になります。

8. コンソールに次のコマンドを入力し、DIOP サーバーを有効化します。

```
load diiop
```

9. 次の場所に diiop_ior.txt が生成されていることを確認します。

```
C: ¥Lotus¥Domino¥Data¥domino¥html¥diiop_ior.txt
```

10. コンソールに次のコマンドを入力し、HTTP サービスを有効化します。

```
load http
```

- 別のサービスがポート 80 を使用している場合、HTTP サービスは開始されません。ポート 80 で実行されているサービスを停止し、コンソールに次のコマンドをもう一度入力します。**load http**

または

- 既存のサービスを使用します。既存のサービスを使用するには、ポート 80 で実行されている Web サーバーのルートまたはホームディレクトリに diiop_ior.txt ファイルをコピーします。HTTP サービスと DIOP サービスの両方を notes.ini ファイルに含め、サーバーの起動時に両方のサービスを確実に開始することができます。

Lotus Notes の設定

Sun Java System Portal Server のメールチャネルとカレンダーチャネルを使用して Lotus Notes システムにアクセスするには、Sun Java System Portal Server に別のファイルを追加する必要があります。これは、NCSO.jar というファイルです。このファイルは、Lotus Notes 製品の CD または IBM の Web サイトから取得する必要があります。

IBM 製の Domino Designer および Domino Server 製品では、domino¥java サブディレクトリに格納されています。また、次の Web サイトからダウンロードすることもできます。

<http://www-10.lotus.com/ldd/toolkits>

「Lotus Domino Toolkit」リンクにアクセスし、Java/Corba R5.0.8 アップデートのリンクにアクセスします。

注 ダウンロードされるファイルは .exe ファイルです。このファイルを含む複数のファイルが解凍されます。

関連する 4 つの Web コンテナに関する後述の説明に従って、Web コンテナ (Web サーバーまたはアプリケーションサーバー) のグローバルクラスパスに NCSO.jar ファイルを配置します。4 つの Web コンテナのうち 3 つでは、NCSO.jar ファイルを /usr/share/lib に配置します。次の表に、これからの手順をまとめます。この表では、NCSO.jar ファイルをシステムクラスパスと Portal WAR のどちらに配置できるかを示し、グローバルクラスパスに JAR ファイルを配置するプロセスを要約しています。また、特別な手順が必要かどうかを示します。必要な場合は、この節の後半で説明します。

Web コンテナ	システムクラスパス	Portal WAR	特別な指示
Sun Java System Web Server	あり	あり	なし
Sun Java System Application Server	あり	あり	なし
BEA WebLogic Server	あり	なし	システムクラスパスをどのように更新するか
IBM WebSphere Application Server	なし	あり	JAR ファイルをどのように切り取るか

次の手順では、Web コンテナに対する管理アクセス権が必要です。また、Web コンテナに関するさまざまな手順やコマンドについて詳細な情報を参照するには、Web コンテナのマニュアルを利用できる必要もあります。Sun Java System Web コンテナについては、『Sun Java System Application Server 管理者ガイド』または『Sun Java System Web Server, Enterprise Edition 管理者ガイド』を参照してください。

[Sun Java System Web Server](#)

[Sun Java System Application Server](#)

[BEA WebLogic Server](#)

[IBM WebSphere Application Server](#)

Sun Java System Web Server

1. 次の Sun Java System Portal Server ディレクトリに NCSO.jar を配置します。
/usr/share/lib
2. 次のファイルが含まれるように、Web コンテナのシステムクラスパスを更新します。
/usr/share/lib/NCSO.jar

- a. Sun Java System Web Server 管理コンソールを起動します。
 - b. Sun Java System Web Server インスタンスを選択します。
 - c. 「Manage」をクリックします。
 - d. 「Java」タブを選択します。
 - e. 「JVM Path Settings」を選択します。
 - f. クラスパスのサフィックスに `/usr/share/lib/NCSO.jar` を追加します。
 - g. 「OK」を選択します。
 - h. 「Apply」を選択します。
3. 必須の操作ではありませんが、Sun Java System Web Server を再起動することをお勧めします。

NCSO.jar ファイルのオプション配置

1. 次のディレクトリに NCSO.jar ファイルを配置します。
`PortalServer-base/SUNWps/web-src/WEB-INF/lib`
2. 次のコマンドを実行して、Web アプリケーションを配備し直します。
`PortalServer-base/SUNWps/bin/deploy redeploy`
3. Web コンテナを再起動します。

Sun Java System Application Server

1. 次の Sun Java System Portal Server ディレクトリに NCSO.jar を配置します。
`/usr/share/lib`
2. Sun Java System Application Server の管理コンソールを使用して、
`/usr/share/lib/NCSO.jar` が含まれるように Web コンテナのクラスパスを更新します。
 - a. Sun Java System Application Server 管理コンソールを起動します。
 - b. ドメインを選択します。
 - c. サーバーインスタンスを選択します。
 - d. サーバーインスタンスの表示で「JVM 設定」タブを選択します。
 - e. 「JVM 設定」タブの下にある「パス設定」を選択します。
 - f. 「クラスパスのサフィックス」リストに `/usr/share/lib/NCSO.jar` を追加します。
 - g. 「保存」をクリックします。
 - h. インスタンスの「一般」タブの下にある「変更を適用」を選択します。

- i. 「再起動」を選択します。

NCSO.jar ファイルのオプション配置

1. 次のディレクトリに NCSO.jar ファイルを配置します。

PortalServer-base/SUNWps/web-src/WEB-INF/lib

2. 次のコマンドを実行して、Web アプリケーションを配備し直します。

PortalServer-base/SUNWps/bin/deploy redeploy

この *PortalServer-base* は、Portal Server が最初にインストールされていた場所を表します。

3. Web コンテナを再起動します。

BEA WebLogic Server

1. 次の Sun Java System Portal Server ディレクトリに NCSO.jar を配置します。

/usr/share/lib

2. コマンドラインを使用して、*/usr/share/lib/NCSO.jar* が含まれるように Web コンテナのクラスパスを更新します。

- a. Web コンテナのインストールディレクトリに移動します。

WebContainer-base/bea/wlserver6.1/config

この *WebContainer-base* は、Web コンテナが最初にインストールされていた場所を表します。

- b. 次のドメインインスタンスが格納されているディレクトリに移動します。

mydomain

- c. 任意のエディタを使用して *startWebLogic.sh* ファイルを編集します。

- d. CLASSPATH の最後に */usr/share/lib/NCSO.jar* を追加します。

注

startWebLogic.sh ファイルに、CLASSPATH 定義が複数含まれている場合があります。CLASSPATH の最後の定義を探し、その末尾に次の文字列を追加してください。

/usr/share/lib/NCSO.jar

- e. Web コンテナを再起動します。

IBM WebSphere Application Server

1. NCSO.jar ファイルの *org/w3c/dom/* と *org/xml/sax/* の下のクラスを切り取り、jar 形式に戻します。

対象となるのは、次のクラスです。

- o org/w3c/dom/Document.class
- o org/w3c/dom/Node.class
- o org/xml/sax/InputSource.class
- o org/xml/sax/SAXException.class

このタスクは、さまざまな方法で実行できます。ここでは、2つの例を取り上げます。適している方法で実行してください。

- o 次の方法では、jar ファイルを手動で展開し、再び jar 形式に戻す必要があります。
 - a. NCSO.jar ファイルをダウンロードし、次のディレクトリに配置します。
/tmp/ncsoprune/work
 - b. このディレクトリ内でファイルを展開します。
 - c. 前述の4つのクラスを消去します。
 - d. ファイルを jar 形式に戻します。
- o 次の方法では、jar 化と jar の展開ロジックを自動化したスクリプトを実行する必要があります。
 - a. NCSO.jar ファイルをダウンロードし、次のディレクトリに配置します。
/tmp/ncsoprune/work
 - b. 次のスクリプトを実行します。

```
#!/bin/ksh
JAR=/usr/j2se/bin/jar
JAR_FILE=NCSO.jar
RM=/usr/bin/rm
BASE_DIR=/tmp/ncsoprune
WORK_DIR=${BASE_DIR}/work
# cd to director of jar file
cd $WORK_DIR
# unjar
$JAR xvf $JAR_FILE
# prune classes
$RM $WORK_DIR/org/w3c/dom/Document.class
$RM $WORK_DIR/org/w3c/dom/Node.class
$RM $WORK_DIR/org/xml/sax/InputSource.class
$RM $WORK_DIR/org/xml/sax/SAXException.class
# jar
$JAR cvf $BASE_DIR/$JAR_FILE META-INF com lotus org
```

2. jar 形式に戻した NCSO.jar ファイルを次のディレクトリに配置します。

PortalServer-base/SUNWps/web-src/WEB-INF/lib

3. 次のコマンドを実行して、Web アプリケーションを配備し直します。

```
PortalServer-base/SUNWps/bin/deploy redeploy
```

この *PortalServer-base* は、Portal Server が最初にインストールされていた場所を表します。

4. Web コンテナを再起動します。

デフォルト組織の下に新規ユーザーを作成する

1. インターネットブラウザで `http://hostname:port/amconsole` (たとえば `http://psserver.company22.example.com:80/amconsole`) と入力し、Sun Java System Access Manager の管理コンソールにログインします。
2. 「アイデンティティ管理」タブをクリックし、ナビゲーションパネルに「表示」ドロップダウンリストを表示します。
3. 「表示」ドロップダウンリストから「ユーザー」を選択し、「ユーザー」ページを表示します。
4. 「新規」をクリックして、データパネルに「新規ユーザー」ページを表示します。
5. ユーザーに割り当てるサービスを選択します。
少なくとも、ポータルデスクトップと SSO アダプタを選択します。
6. ユーザーに関する情報を入力します。
7. 「作成」をクリックします。
ナビゲーションパネルの「ユーザー」リストに新しいユーザーの名前が表示されます。

HTTPS が有効な Messaging Server とメールプロバイダが連携するように設定する

メールチャネルは、自動的に HTTP プロトコルをサポートし、より安全な HTTPS プロトコルを自動的にサポートしません。しかし、Sun Java System Messaging Server で HTTPS が有効な場合は、ここで説明する手順を実行して、Sun Java System Messaging Server と正常に連携するようにメールプロバイダを設定することができます。これらの手順は、Microsoft Exchange Server と IBM Lotus Notes サーバーには適用されません。

Web コンテナに関する情報と注意点

HTTPS が有効な Sun Java System Messaging Server に合わせてメールプロバイダを設定する場合、使用する Web コンテナの種類 (Sun Java System Web Server、Sun Java System Application Server、BEA WebLogic Server、または IBM WebSphere Application Server) に応じて Web コンテナに関する手順が異なります。使用する Web コンテナの種類に関係なく、Web コンテナに対する管理権限は必要です。また、信頼データベースの初期化、証明書の追加、Web コンテナの再起動について詳細な情報を参照するには、Web コンテナのマニュアルを利用できる必要もあります。Sun Java System Web コンテナに関する上記タスク、およびセキュリティ関連の問題については、『Sun Java System Application Server Administrator's Guide to Security』または『Sun Java System Web Server, Enterprise Edition 管理者ガイド』を参照してください。

HTTPS が有効な Messaging Server とメールプロバイダが連携するように設定する

1. Sun Java System Portal Server を実行する Web コンテナの信頼データベースを初期化します。詳細については、前述の適切なマニュアルを参照してください。
2. 信頼されている認証局 (TCA) の SSL 証明書がインストールされていない場合は、インストールします。
3. 必須の操作ではありませんが、Web コンテナを再起動することをお勧めします。
4. HTTPS 専用の新しい SSO アダプタテンプレートを追加します。この例で使用するテンプレートの名前は、SSL セキュリティプロトコルの使用を明示した SUN-ONE-MAIL-SSL です。

注

SSO アダプタテンプレートと、関連する SSO アダプタ設定は、さまざまな方法で設定できます。次に示す手順は、代表的な設定方法を示しています。既存のテンプレートと設定を編集するより安全性が高いため、ここでは、テンプレートと設定を新たに作成する手順を説明します。

編集のほうに適していると考えられる場合は、編集を行ってください。ただし、編集時に SSO アダプタテンプレートと SSO アダプタ設定の名前を変更した場合は、メールチャネルのプロパティを編集して、SSO アダプタ名を変更する必要があります。

SSO アダプタテンプレートまたは SSO アダプタ設定で編集が必要なのは、次の 2 つの項目です。

- `clientProtocol`
- `clientport`

新しい SSO アダプタテンプレートを作成するこの例では、`clientProtocol` 属性を `default` 属性として設定します。このため、これは SSO アダプタ設定には表示されず、SSO アダプタテンプレートに表示されます。`clientProtocol` 属性は、`http` から `https` に変更する必要があります。この属性の編集後のテンプレートコードは、次のようになります。

```
clientProtocol=https
```

この例では、`clientPort` 属性を `merge` 属性として設定します。このため、これは SSO アダプタ設定に表示されます ([421 ページの手順 5](#) を参照)。`clientPort` 属性を `default` 属性として設定した場合は、これは SSO アダプタテンプレートに表示されます。クライアントポートは、HTTPS 専用に予約されているポートに変更する必要があります。ここでは、HTTPS プロトコルがデフォルトで使用するポート番号である、ポート 443 を使用します。この属性の編集後のテンプレートコードは、次のようになります。

```
&clientPort=443
```

-
- インターネットブラウザで `http://hostname:port/amconsole` (たとえば `http://psserver.company22.example.com:80/amconsole`) と入力し、Sun Java System Access Manager の管理コンソールにログインします。
 - 「サービス設定」タブをクリックします。設定できるサービスのリストがナビゲーションパネルに表示されます。
 - SSO アダプタの横の矢印をクリックして、データパネルに「SSO アダプタ」ページを開きます。
 - テンプレート名を入力し、メニューから既存のテンプレートを選択します。
 - 「次へ」をクリックします。
 - 「テンプレートプロパティ」ページが表示されます。

- g. 必要に応じてプロパティを修正します。

コード例 17-4 は、参照用の一般的な設定例です。実際に入力するテンプレートは、多くの場合は異なる情報を含んでいます。たとえば、SUN-ONE-MAIL-SSL という名前を使用しない限り、configName プロパティタイプの値は異なります。さらに、属性に指定する default と merge の設定も、サイト要件によってこの例とは異なります。

- h. 完了したら、「保存」を選択します。

コード例 17-4 HTTPS Messaging Server 用のメール SSO アダプタテンプレート

```
default|imap:///?configName=SUN-ONE-MAIL-SSL
  &encoded=password
  &default=protocol
  &default=clientProtocol
  &default=type
  &default=subType
  &default=enableProxyAuth
  &default=proxyAdminUid
  &default=proxyAdminPassword
  &default=ssoClassName
  &merge=host
  &merge=port
  &merge=uid
  &merge=password
&merge=smtpServer
  &merge=clientPort
  &clientProtocol=https
  &enableProxyAuth=false
  &proxyAdminUid=[PROXY-ADMIN-UID]
  &proxyAdminPassword=[PROXY-ADMIN_PASSWORD]
  &type=MAIL-TYPE
  &subType=sun-one
  &ssoClassName=com.sun.ssoadapter.impl.JavaMailSSOAdapter
  &default=enablePerRequestConnection
  &enablePerRequestConnection=false
```

この時点で、IMAP プロトコルから始まる文字列が複数存在する可能性があります。これは、問題ありません。

5. HTTPS 専用の新しい SSO アダプタ設定を追加します。この例で使用する設定の名前は、対応する SSO アダプタテンプレートに合わせて sunOneMailSS1 とします。

注

419 ページの手順 4 で示した注意を参照してください。

- a. インターネットブラウザで `http://hostname:port/amconsole` (たとえば `http://psserver.company22.example.com:80/amconsole`) と入力し、Sun Java System Access Manager の管理コンソールにログインします。
 - b. 「アイデンティティ管理」タブをクリックし、ナビゲーションパネルに「表示」ドロップダウンリストを表示します。
 - c. 「表示」ドロップダウンリストの「サービス」をクリックします。
 - d. 「シングルサインオンアダプタの設定」という見出しが表示されるまでナビゲーションパネルをスクロールし、SSO アダプタ項目の隣にある矢印をクリックします。データパネルに「SSO アダプタ」ページが表示されます。
 - e. 設定を説明するフィールド(「追加」ボタンと「削除」ボタンのすぐ上にある)が空の状態、その内側をクリックします。
 - f. 「SSO アダプタの設定」から「新規」をクリックして、SSO アダプタの設定を追加します。
 - g. 「新規設定」ページが表示されます。
 - h. 設定名を入力し、メニューから SSO アダプタテンプレートを選択します。
 - i. 「次へ」をクリックします。
 - j. 「設定プロパティ」ページが表示されます。
 - k. 必要に応じてプロパティを修正します。
 - l. 完了したら、「保存」を選択します。
6. ポータルデスクトップに新しいメールチャネルを追加します。

手順 4 と手順 5 では、SSO アダプタテンプレートと SSO アダプタ設定を新規作成して新規チャネルを作成する方法について説明しました。この手順では、エンドユーザーがこのチャネルを利用できるように設定します。

新しいチャネルに名前を付ける条件は、明的なものであるということに過ぎません。ここでは、SunOneMailSSLChannel という名前を使用します。

- a. インターネットブラウザで `http://hostname:port/amconsole` (たとえば `http://psserver.company22.example.com:80/amconsole`) と入力し、Sun Java System Access Manager の管理コンソールにログインします。
- b. 「アイデンティティ管理」タブをクリックし、ナビゲーションパネルに「表示」ドロップダウンリストを表示します。
- c. 「表示」ドロップダウンリストから「サービス」を選択し、設定可能なサービスのリストを表示します。
- d. 「Portal Server 設定」の見出しの下で、「ポータルデスクトップ」の隣にある矢印をクリックします。データパネルに「ポータルデスクトップ」ページが表示されます。

- e. 必要に応じてスクロールし、「チャネルおよびコンテナを管理」リンクをクリックします。
- f. 「チャネル」という見出しまでスクロールし、「新規」をクリックします。
- g. 「チャネル名」フィールドに新しいチャネルのサイト名を入力します。たとえば、SunJavaMailSSLChannel のように指定します。
- h. 「プロバイダ」ドロップダウンメニューから「MailProvider」を選択します。
- i. 「了解」をクリックします。「チャネルおよびコンテナを管理」Web ページが表示され、新たに作成したチャネルも表示されます。
- j. 「チャネル」の見出しまでスクロールし、作成したチャネルの名前(この例では SunOneMailSSLChannel)の隣にある「プロパティの編集」をクリックします。
- k. タイトルのフィールドまでスクロールし、表示されているテキスト(mail など)を選択、削除して、プロバイダのタイトルを入力します。たとえば、SSL Mail Account のように指定します。
- l. 説明のフィールドに表示されているテキスト(mail など)を選択して削除し、プロバイダの説明を入力します。ここでは、前述のサブステップでタイトルの例として使用したのと同じ SSL Mail Account を説明の例に使用します。
- m. ページを下にスクロールし、「SSO アダプタ」に表示されているテキスト(sunOneMail など)を選択、削除し、[421 ページの手順 5](#) で使用した SSO アダプタ設定と同じ名前(この例では sunOneMailSSL)を入力します。
- n. さらに下にスクロールし、「保存」をクリックします。
- o. ページを上スクロールし、Container Path という句の後の最初の単語である top をクリックします。
- p. 「コンテナチャネル」という見出しまで下にスクロールし、新しいチャネルを追加するコンテナのリンクをクリックします。たとえば、MyFrontPageTabPanelContainer をクリックします。対応する「プロパティの編集」リンクをクリックしないように注意してください。
- q. 「チャネル管理」という見出しまで下にスクロールし、必要に応じて「使用可能」フレーム内でもスクロールして、新たに作成したチャネルの名前をクリックして選択します。

この例では、チャネル名は SunOneMailSSLChannel です。
- r. 「コンテンツページでエンドユーザーが使用可能」リストまたは「ポータルデスクトップで表示可能」リストにチャネルを追加します。

チャネルを追加するリストの上にある「追加」ボタンをクリックします。

- s. ページを上スクロールし、「チャンネル管理」という見出しの下の「保存」をクリックします。

これで、HTTPS が有効な Messaging Server にログインし、使用できるようになります。

Portal Server システムの管理

この章では、Sun Java™ System Portal Server システムの維持に関連した、さまざまな管理タスクについて説明します。

この章で説明する内容は次のとおりです。

- [Secure Sockets Layer \(SSL\) の設定](#)
- [Portal Server の設定のバックアップと復元](#)
- [複数の UI ノードインストールの管理](#)
- [HTTP プロキシを使用する Portal Server のインスタンスの設定](#)
- [Portal Server ログの管理](#)
- [Portal Server のデバッグ](#)

Secure Sockets Layer (SSL) の設定

次の方法によって、Portal Server および関連するコンポーネントで Secure Sockets Layer (SSL) を設定することができます。

- **Portal Server:** ゲートウェイではなく Portal Server システムだけに SSL を設定すると、イントラネットは「公開」されます。

Portal Server ユーザーインタフェースノード (Sun Java™ System Access Manager 管理コンソール、デスクトップ、サーブレットなどが実行されるノード) とゲートウェイノードの間、および Portal Server ユーザーインタフェースノードとエンドユーザーコンピュータの間で SSL を使用することができます。

- **Sun Java™ System Directory Server:** Sun Java System Directory Server の SSL を設定し、Sun Java System Access Manager と Portal Server の間でセキュリティ保護された接続を使用できます。Directory Server で SSL を有効にする方法の詳細については、次の URL の『Sun Java System Access Manager Installation and Configuration Guide』の第 6 章「Basic Configurations」を参照してください。

<http://docs.sun.com/source/816-5626-10/contents.html>

注 Directory Server で SSL を設定した場合、SSL を無効にしてから、Portal Server インストールスクリプトとともに Directory Server をアンインストールしてください。さらに、コマンドラインで `dpsadmin` コマンドを使用する場合も、SSL を無効にする必要があります。

- Sun Java™ System Portal Server:Secure Remote Access: ゲートウェイの SSL を設定すると、イントラネットはセキュリティ保護されます。ゲートウェイで SSL を設定する手順については、『Sun Java System Portal Server Secure Remote Access 6 2005Q1 管理ガイド』を参照してください。

Portal Server で SSL を設定する

Portal Server のインストール中にマシンで SSL を動作させる場合は、この手順を使用します。

1. Portal Server をインストールした Web サーバーの信頼データベースを作成します。

詳細については、次の URL の『Sun Java System Web Server 6 2004Q2, Enterprise Edition Administration Guide』の第 5 章「Creating a Trust Database」を参照してください。

<http://docs.sun.com/source/816-2140-01/contents.htm>

2. Portal Server をインストールした Web サーバーの証明書を要求し、Web サーバーのインスタンスに証明書をインストールします。

詳細については、『Sun Java System Web Server 6 2004Q2, Enterprise Edition Administration Guide』の第 5 章「Requesting and Installing a VeriSign Certificate」または「Requesting and Installing Other Server Certificates」を参照してください。

3. Portal Server の Web サーバーインスタンスのために暗号化を有効にします。

Web サーバー管理コンソールの「Configuration」タブで、「Add Listen Socket」を選択して「Edit Listen Socket」を選択し、セキュリティをオンにします。

詳細については、『Sun Java System Web Server 6 2004Q2, Enterprise Edition Administration Guide』の第 5 章「Turning Security On」を参照してください。

4. Web サーバーの管理コンソールで「Apply」と「Apply Changes」をクリックします。
5. Web コンテナを再起動します。

Web コンテナの起動手順については、使用する Web コンテナのマニュアルを参照してください。

6. 証明書データベースを取得するためのパスワードの入力が求められます。

注 再起動するたびにパスフレーズを入力することを回避するため、Web サーバーのパスフレーズを含む `.wtpass` という名前のファイルを作成し、`AccessManager-base/SUNWam/config` ディレクトリに配置します。このファイルを使用せずにセキュリティ保護された Web サーバーのシステムをリブートした場合は、システムコンソールでパスフレーズを入力する必要があります。

7. 次の方法で、SSL を使用して Portal Server にログオンできることを確認します。
 - Sun Java System Access Manager の管理コンソールにログオンするには、次のように入力します。
`https://server:port/amconsole`
 - デスクトップにユーザーとしてログオンするには、次のように入力します。
`https://server:port/deploy_uri`

たとえば、次のようになります。
`https://sesta:80/portal/dt`

SSL を使用する既存の Portal Server インストールを変更する

Portal Server のインストール時に「Do you want to run SSL on *hostname*?」という質問に対して、「**n**」を入力した場合は、この手順を使用します。詳細は、『Sun Java System Portal Server 6 2004Q2 Installation Guide』を参照してください。

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. プラットフォームの隣にあるプロパティの矢印をクリックします。
プラットフォーム属性がデータパネルに表示されます。
4. サーバーリストで、`http` を `https` に変更します。
5. 「保存」をクリックして、変更を保存します。

6. Web サーバーで証明書をインストールします。
詳細は、[426 ページの「Portal Server で SSL を設定する」](#)の[手順 1](#)～[手順 4](#)を参照してください。
7. `server.xml` および `magnus.conf` ファイルを
`/AccessManager-base/SUNWam/servers/https-hostname-domain/conf_bk` ディレクトリから、`/AccessManager-base/SUNWam/servers/https-hostname-domain/config` ディレクトリへコピーします。
8. 証明書のためのルート CA がインストールされていない場合は、
`/AccessManager-base/SUNWam/lib/AMConfig.properties` ファイルに、次の行を追加します。

```
com.sun.am.jssproxy.trustAllServerCerts=true
```

このオプションは、証明書が信用できることを JSS に示します。
9. `/AccessManager-base/SUNWam/lib/AMConfig.properties` ファイルで、次の URL の
`http` を `https` に変更します。

```
com.sun.am.server.protocol  
com.sun.am.naming.url  
com.sun.am.notification.url  
com.sun.am.session.server.protocol  
com.sun.services.cdsso.CDCURL  
com.sun.services.cdc.authLoginUrl
```
10. Web コンテナを再起動します。
11. 証明書データベースを取得するためのパスワードの入力が求められます。
詳細は、『[Sun Java System Directory Server 管理ガイド](#)』の第 11 章「SSL の管理」を参照してください。

SSL を使用するように Portal Server インスタンスを設定する

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
3. プラットフォームの隣にあるプロパティの矢印をクリックします。

プラットフォーム属性がデータパネルに表示されます。

4. サーバリストで、http を https に変更します。
5. 「保存」をクリックして、変更を保存します。
6. Web サーバーで証明書をインストールします。

詳細は、[426 ページの「Portal Server で SSL を設定する」](#)の手順 1 ～手順 4 を参照してください。

7. このサーバーが複数のインスタンスの一部である場合、server.xml および magnus.conf ファイルを
/AccessManager-base/SUNWam/servers/https-instance_nickname/conf_bk ディレクトリから /AccessManager-base/SUNWam/servers/https-instance_nickname/config ディレクトリへコピーします。

8. 証明書のためのルート CA がインストールされていない場合は、
/AccessManager-base/SUNWam/lib/AMConfig-instance_nickname.properties ファイルに次の行を追加します。

```
com.sun.am.jssproxy.trustAllServerCerts=true
```

このオプションは、証明書が信用できることを JSS に示します。

9. /AccessManager-base/SUNWam/lib/AMConfig-instance_nickname.properties ファイルで、次の URL の http を https に変更します。

```
com.sun.am.server.protocol
```

```
com.sun.am.naming.url
```

```
com.sun.am.notification.url
```

```
com.sun.am.session.server.protocol
```

```
com.sun.services.cdsso.CDCURL
```

```
com.sun.services.cdc.authLoginUrl
```

10. Web コンテナを再起動します。

11. 証明書データベースを取得するためのパスワードの入力が求められます。

詳細は、『Sun Java System Directory Server 管理ガイド』の第 11 章「SSL の管理」を参照してください。

Portal Server の設定のバックアップと復元

Portal Server のユーザーおよびサービス設定は、Directory Server の LDAP ディレクトリ情報ツリー (DIT) に保存されます。これにより LDIF (Lightweight Directory Interchange Format) ファイルによって、設定情報のバックアップと復元を行えます。

Portal Server の設定をバックアップする

Portal Server の設定情報のバックアップを作成するには、db2ldif コマンドを使用します。このコマンドは、Directory Server のベースディレクトリ内にある `slapd-hostname` ディレクトリで使用できます。たとえば、サーバー `sesta` のデフォルトのインストールディレクトリ (`/usr/ldap`) に Directory Server がインストールされた場合、ベースディレクトリは `/usr/ldap/slapd-sesta` になります。

1. ディレクトリを db2ldif コマンドを含む Directory Server のベースディレクトリに変更します。
2. Portal Server の DIT の最上位レベルを指定する `-s` オプションを付けて db2ldif コマンドを使用し、LDIF ファイルに設定を保存します。たとえば、DIT の最上位レベルが `isp` である設定を保存するには、次のように入力します。

```
cd DirectoryServer-base/slapd-HOSTNAME
```

```
./db2ldif -s "o=isp"
```

データは LDIF ファイルに保存されます。コマンドは現在のディレクトリにファイルを保存します。ファイル名の形式は次のとおりです。

```
YYYY_MM_DD_HHMMSS.ldif
```

ファイルが保存されたあと、出力表示の例は次のようになります。

```
[16/May/2002:14:11:25 -0700] - Backend Instance:userRoot
ldiffile:/usr/ldap/slapd-sesta/ldif/2002_05_16_141122.ldif
[16/May/2002:14:11:28 -0700] - export userRoot:Processed 178 entries (100%)
```

Portal Server の設定を復元する

ldif2db コマンドを使用して、db2ldif コマンドによってバックアップを作成した Portal Server の設定情報を復元することができます。このコマンドは、Directory Server のベースディレクトリ内にある slapd-hostname ディレクトリで使用できます。たとえば、サーバー sesta のデフォルトのインストールディレクトリ (/usr/ldap) に Directory Server がインストールされた場合、ベースディレクトリは /usr/ldap/slapd-sesta になります。

1. 次のように入力して、ディレクトリを ldif2db コマンドを含む Directory Server のベースディレクトリに変更します。

```
cd DirectoryServer-base/slapd-HOSTNAME
```

2. 次のように入力して、Directory Server を終了します。

```
./stop-slapd
```

3. Portal Server の DIT の最上位レベルを指定する -s オプション、およびファイル名を指定する -i オプションを付けて ldif2db コマンドを使用し、LDIF ファイルから Directory Server へ設定を復元します。たとえば、前の手順で保存された LDIF ファイルを isp の DIT の最上位レベルへ復元するには、次のように入力します。

```
./ldif2db -s "o=isp" -i
/usr/ldap/slapd-sesta/ldif/2002_05_16_141122.ldif
```

ファイルが復元されたあと、出力表示の例は次のようになります。

```
importing data ...

[16/May/2002:16:37:02 -0700] - Backend Instance:userRoot

[16/May/2002:16:37:03 -0700] - import userRoot:Index buffering
enabled with bucket size 13

[16/May/2002:16:37:03 -0700] - import userRoot:Beginning import
job...

[16/May/2002:16:37:03 -0700] - import userRoot:Processing file
"/usr/ldap/slapd-sesta/ldif/2002_05_16_141122.ldif"

[16/May/2002:16:37:04 -0700] - import userRoot:Finished scanning
file "/usr/ldap/slapd-sesta/ldif/2002_05_16_141122.ldif" (178
entries)

[16/May/2002:16:37:05 -0700] - import userRoot:Workers finished;
cleaning up...

[16/May/2002:16:37:08 -0700] - import userRoot:Workers cleaned up.

[16/May/2002:16:37:08 -0700] - import userRoot:Cleaning up producer
thread...
```

```
[16/May/2002:16:37:08 -0700] - import userRoot: Indexing complete.  
Post-processing...  
[16/May/2002:16:37:08 -0700] - import userRoot:Flushing caches...  
[16/May/2002:16:37:08 -0700] - import userRoot:Closing files...  
[16/May/2002:16:37:09 -0700] - import userRoot: Import complete.  
Processed 178 entries in 6 seconds. (29.67 entries/sec)
```

4. 次のように入力して、Directory Server を再起動します。

```
./start-slapped
```

Portal Server のネットワーク設定の変更

Portal Server ソフトウェアを実行するサーバーをネットワーク間で物理的に移動する場合、`/etc/hosts` ファイルで IP アドレスをマッピングする完全修飾ドメイン名のみを変更する必要があります。変更する必要のあるハードコードされたアドレスは他にありません。

複数の UI ノードインストールの管理

複数の UI ノードに Portal Server ソフトウェアをインストールする場合は、Sun Java System Access Manager 管理コンソールでプラットフォームの属性に設定の変更を加える必要があります。「サーバーリスト」属性を編集して、それぞれの UI ノードに URL を組み込みます。

Sun Java System Access Manager ネーミングサービスは、初期化時に「サーバーリスト」属性を読み込みます。このリストは、単一の Sun Java System Access Manager 設定に Sun Java System Access Manager セッションサーバーを含んでいます。たとえば、2つの Sun Java System Access Manager サーバーがインストールされ、1つのサーバーとして動作する場合、それらは両方ともこのリストに含まれる必要があります。サービス URL の要求で指定されたホストがこのリストにない場合、ネーミングサービスは要求を拒否します。リストの最初の値は、ホスト名およびインストール中に指定されたサーバーのポートを指定します。追加のサーバーは、`protocol://server:port` という形式で追加できます。

サーバーリストに Portal Server を追加する

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
グローバルサービスがナビゲーションパネルに表示されます。
3. プラットフォームの隣にあるプロパティの矢印をクリックします。
プラットフォーム属性がデータパネルに表示されます。
4. 「サーバーリスト」属性を編集します。
UI ノードとして機能しているサーバーごとに、`http://host1.sesta.com:80` などのサーバー URL を入力し、「追加」ボタンを選択します。「サーバーリスト」に URL が表示されます。
5. 「保存」をクリックします。
6. Web コンテナを再起動します。

HTTP プロキシを使用する Portal Server のインスタンスの設定

Portal Server ソフトウェアを、インターネットまたはイントラネットの特定の部分に直接アクセスできないホストにインストールした場合は、HTTP プロキシを使用するようにインスタンスを設定できます。

Portal Server は、Portal Server Web アプリケーションを実行している Web コンテナで `http.proxyHost` と `http.proxyPort` の Java 仮想マシン (JVM) システムプロパティを設定することにより、HTTP プロキシを使用するように設定されます。JVM システムプロパティの設定方法は、Web コンテナによって異なります。このセクションで説明される手順は、特に Sun Java System Web Server のインスタンスを設定して HTTP プロキシを使用するためのものです。

1. 次のように入力して、ディレクトリを、インスタンスの設定を含む Web Server のベースディレクトリに変更します。

```
cd /WebServer-base/SUNWam/servers/https-hostname-domain/config
```

2. このディレクトリ内の `server.xml` を編集し、次の行を追加します。

```
<JVMOPTIONS>-Dhttp.proxyHost=proxy_host</JVMOPTIONS>
```

```
<JVMOPTIONS>-Dhttp.proxyPort=proxy_port</JVMOPTIONS>
```

ここで、*proxy_host* は、プロキシホストの完全修飾ドメイン名、*proxy_port* は、プロキシが動作しているポートです。

注 server.xml ファイルにプロキシ設定 (`http.proxyHost=` と `http.proxyPort=` オプションを使用) がある場合は、`http.nonProxyHosts=proxy_host` オプションを追加することもできます。プロキシサーバーのアクセスリストに Portal Server が追加されていない場合、プロキシサーバー経由で Portal Server にアクセスすることはできません。

Portal Server ログの管理

Portal Server のロギングを設定して、フラットファイルまたはデータベースにログ情報を記録することができます。データベースにログを記録する場合、JDBC プロトコルが使用されます。

ファイルにログを記録するように設定する

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
グローバルサービスがナビゲーションパネルに表示されます。
3. 「ログ」の隣にあるプロパティの矢印をクリックします。
データパネルに「ログ」属性が表示されます。
4. 「ログタイプ」属性としてファイルを選択します。
5. 「ログの場所」属性にログファイルのディレクトリパスを指定します。
6. 「最大ログサイズ」属性に、ログファイルの最大サイズをバイト単位で指定します。
7. 「履歴ファイルの数」属性に、バックアップログの数を指定します。
8. 「保存」をクリックします。

データベースにログを記録するように設定する

1. Sun Java System Access Manager 管理コンソールに、管理者としてログインします。
デフォルトでは、ロケーションパネルでは「アイデンティティ管理」が選択され、ナビゲーションパネルには作成済みのすべての組織が表示されています。
2. ロケーションパネルで「サービス設定」を選択します。
グローバルサービスがナビゲーションパネルに表示されます。
3. ナビゲーションパネルの「ログ」サービスの横にあるプロパティの矢印をクリックします。
データパネルに「ログ」属性が表示されます。
4. 「ログタイプ」属性として DB を選択します。
5. 「データベースユーザー名」および「データベースユーザーパスワード」属性に、データベースに接続するユーザー名とパスワードを指定します。
6. 「データベースドライバ名」属性に、ロギングに使用するドライバを指定します。
7. 「保存」をクリックします。

Portal Server のデバッグ

ここでは、デバッグのレベルを設定して、Portal Server のさまざまなコンポーネントの問題を解決する方法を説明します。

Sun Java System Access Manager のデバッグレベルを設定する

デバッグのレベルにより、デバッグログに送信されたメッセージの種類を定義することができます。次のレベルを使用できます。

- **off:** メッセージはデバッグログに送信されません。
- **error:** エラーメッセージがデバッグログに送信されます。
- **warning:** 警告およびエラーメッセージがデバッグログに送信されます。
- **message:** ステータス、警告およびエラーメッセージがデバッグログに送信されます。

デフォルトでは、デバッグメッセージは `/var/opt/SUNWam/debug` ディレクトリのログファイルに送信されます。

デバッグのレベルを設定するには、次の手順を実行します。

1. `/etc/opt/SUNWps/desktopconfig.properties` ファイルの次の行にデバッグレベルを定義します。

```
debugLevel=value
```

2. Web コンテナを再起動します。
3. Sun Java System Web Server のログファイルと、`/var/opt/SUNWam/debug` に格納されているさまざまなログファイルを調べます。

Sun Java System Portal Server のチューニング

第 19 章 「Portal Server のチューニング」

Portal Server のチューニング

この章では、パラメータを設定して Sun Java™ System Portal Server のパフォーマンスと能力を最適化する方法について説明します。Portal Server にバンドルされた *PortalServer-base/SUNWps/bin* ディレクトリの `perftune` スクリプトにより、この章で説明するチューニングプロセスのほとんどが自動的に実行されます。

今回リリースされた `perftune` スクリプトでは、次の点が更新されました。

- 十分なメモリのないシステムがスクリプトによってチューニングされるのを防止する安全手段。
- 別々のマシンにインストールされた Portal Server インスタンスと Access Manager インスタンスをチューニング可能。
- 新しい JVM™ チューニングパラメータ。

注 別々のインスタンスで Portal Server または Access Manager の一方をチューニングする場合は、そのシステムにインストールされているコンポーネントをチューニングするオプションのみを選択してください。たとえば、Portal Server インスタンスだけがインストールされているマシンで `perftune` スクリプトを実行する場合は、Access Manager をチューニングしないでください。

紹介

perftune スクリプトには次の特長があります。

- Solaris™ オペレーティングシステムのカーネルおよび TCP 設定をチューニング (「[Solaris のチューニング](#)」参照)
- 次の製品の一部となっている設定ファイルの変更
 - [Sun Java System Web Server 6.1 のチューニング](#):
 - `WebServer-base/SUNWwbsvr/webserver-instance/config/magnus.conf`
 - `WebServer-base/SUNWwbsvr/webserver-instance/config/server.xml`
 - [Sun Java System Application Server 7.0 のチューニング](#):
 - `Deploy_Domain/Deploy_Instance/config/init.conf`
 - `Deploy_Domain/Deploy_Instance/config/server.xml`
 - `Deploy_Domain/Deploy_Instance/config/server.policy`

`Deploy_Domain` = Application Server のドメインディレクトリおよびアプリケーションサーバードメイン。たとえば、`/var/opt/SUNWappserver7/domains/domain1`

`Deploy_Instance` = Application Server インスタンス。たとえば、`server1`
 - [Sun Java System Directory Server のチューニング](#):
 - `/var/opt/mps/serverroot/slapd-hostname/config/dse.ldif`
 - [Sun Java System Access Manager のチューニング](#):
 - `/etc/opt/SUNWam/config/serverconfig.xml`
 - `/etc/opt/SUNWam/config/AMConfig*properties`
 - [Portal Server デスクトップのチューニング](#)
 - `/etc/opt/SUNWps/desktop/desktopconfig.properties`
- Portal Server デスクトップサービスおよび Sun Java™ System Access Manager 認証サービスのプロパティを変更します。

チューニング方法

perftune スクリプトを実行するには、次の手順を実行します。

1. スーパーユーザーとしてマシンにログインします。
perftune スクリプトを実行するには root アクセスが必要です。
2. ディレクトリを、*PortalServer-base/SUNWps/bin* に変更します。
3. 次のように入力します。

```
./perftune
```

perftune スクリプトは、チューニングプロセス中にサーバーの起動と停止を実行します。また、*filename-orig-date-pid* の形式で変更されたファイルのバックアップコピーを作成します。スクリプトの実行後はシステムを再起動し、チューニングによる変更を有効にします。

Solaris のチューニング

カーネルのチューニング

スクリプトにより、*/etc/system* ファイルに次の設定が追加されます。

- ファイル記述子の制限: 開くファイルの制限数
 - `set rlim_fd_max=16384`
 - `set rlim_fd_cur=16384`
- ストリームキューのサイズ: 送信先ストリームキューによって QFULL が生成される前の syncq (メッセージ数) の深さ
 - `set sq_max_size=0`
- TCP 接続ハッシュサイズ (<= ファイル記述子)
 - `set tcp:tcp_conn_hash_size=8192`

最初のファイル (*/etc/system*) は、次の形式でファイルにコピー

```
/etc/system-orig-'$DATE+%y%m%d'-$$_$
```

注 ほとんどのファイルに対してバックアップが実行されます。スクリプトにより、変更されたファイルのバックアップコピーがそれぞれのディレクトリに次の形式で作成されます。 *filename-orig-date-pid*

TCP パラメータのチューニング

/dev/tcp にある TCP パラメータ (括弧で表記) に次の変更を加えます。

- TCP Time Wait Interval (tcp_time_wait_interval): TCP ソケットが、接続が閉じられた後に TIME_WAIT 状態に留まる時間を 60000 に設定します。
- TCP Fin Wait 2 Interval (tcp_fin_wait_2_flush_interval): TCP ソケットが、接続が閉じられた後に FIN_WAIT_2 状態に留まる時間を 67500 に設定します。
- TCP Maximum Connection Size (tcp_conn_req_max_q): 完全に確立された接続の最大数を 8192 に設定します。
- TCP List Queue (tcp_conn_req_max_q0): 未確立の接続が入っているキューのサイズを 8192 に設定します。
- TCP Packet Drop Time (tcp_ip_abort_interval): パケットがドロップされるまでの時間を 60000 に設定します。
- TCP キープアライブ間隔 (tcp_keepalive_interval): 90000 に設定します。
- TCP Maximum Retransmit Interval (tcp_rexmit_interval_max): 6000 に設定します。
- TCP Minimum Retransmit Interval (tcp_rexmit_interval_min): 3000 に設定します。
- TCP Initial Retransmit Interval (tcp_rexmit_interval_initial): 500 に設定します。
- TCP Smallest Anonymous Port (tcp_smallest_anon_port): 1024 に設定します。
- TCP Initial Packets for Slow Start Algorithm (tcp_slow_start_initial): 2 に設定します。
- TCP Transmit/Receive Buffer Size Limit (tcp_xmit_hiwat および tcp_recv_hiwat): それぞれ 32768 に設定します。

システムの起動時に ndd コマンドを自動的に実行させるため、perftune スクリプトは S99nnds_tcp ファイルを /etc/rc2.d/ ディレクトリにコピーします。

Sun Java System Access Manager のチューニング

Directory Server の接続プール

/etc/opt/SUNWam/config/serverconfig.xml ファイルに次の変更を加えます。

- 最小接続プールサイズを 10 に増やします。
- 最大接続プールサイズを 90 に増やします。

注 /etc/opt/SUNWam/config/serverconfig.xml ファイルには次の形式でバックアップが作成されます。

filename-orig-date-pid

LDAP 認証サービス

- LDAP 接続プールのデフォルトサイズ (最小 : 最大) を 10 : 90 に更新します。

Sun Java System Access Manager サービス設定パラメータ

/etc/opt/SUNWam/config/AMConfig.properties ファイルに次の変更を加えます。

- com.iplanet.am.logstatus を INACTIVE に指定します。
- 予期される同時セッション数が com.iplanet.am.session.maxSession (デフォルトは 50000) の値を超える場合は、com.iplanet.am.session.maxSession の値を増やします。
- com.iplanet.am.session.httpSession.enabled を無効に設定します。
- com.iplanet.am.sdk.cache.maxSize=DSAME_MAX_CACHE_SIZE を指定します。ここで、DSAME_MAX_CACHE_SIZE は、Access Manager と Portal Server のチューニングガイドの推奨設定に基づきます。
MAX_CONCURRENT_SESSIONS=7000、および services registered=3 (出荷時のデフォルト設定) で、DSAME_MAX_CACHE_SIZE=(MAX_CONCURRENT_SESSIONS) * (2 + services registered) を指定します。
- com.iplanet.am.stats.interval の値を 60 に指定します。
- com.iplanet.am.session.purgedelay の値を 5 に指定します。
- com.iplanet.services.stats.state の値を file に指定します。
- com.iplanet.services.states.directory の値を /var/opt/SUNWam/debug に指定します。

Access Manager と Portal Server が別々のマシンにインストールされている場合は、ポーリングモードが有効になります。ポーリングモードでは次のオプションを使用できます。

- ポーリングモードの指定オプション
- ポーリングモードの間隔設定オプション

/opt/SUNWam/lib/AMConfig.properties ファイルのスレッドプールのプロパティは、Portal Server 6 で次のように公表されています。

- `com.ipplanet.am.notification.threadpool.threshold`。このプロパティは、スレッドプールのタスクキューの最大サイズを示します。スレッドプールは、キューにある未処理タスクの数がしきい値を超えると、それ以上の要求を拒否します。しきい値は、システムのメモリリソースの容量によって異なります。それぞれのタスクではおよそ 3K の容量を必要とします。キューに保持可能なタスクの数は、スレッドプールのサイズを考慮して決定します。タスクは、プールにスレッドがない場合にだけキューに送られます。

デフォルト値は 200 に設定されます。特定の用途にとってこの値が大きすぎる場合は、調節できます。たとえば、4 CPU 構成の Ultra Sparc II または III マシンの場合は、値を 40 に設定します。

- `com.ipplanet.am.notification.threadpool.size`。このパラメータにより、負荷が大きい状況でも、Sun Java™ System Application Server での Portal Server の認証の信頼性を高めることができます。デフォルトでは 50 に設定されますが、変更可能です。たとえば、4 CPU 構成の Ultra Sparc II または III マシンの場合は、値を 50 に設定します。

Sun Java System Directory Server のチューニング

Sun Java™ System Directory Server を他のアプリケーションと共有している場合は、設定されているパラメータが他のアプリケーションのパラメータチューニングと競合していないかどうかを検証する必要があります。

/tmp/slapd-DSinstance1 用に十分な仮想メモリ空間を確保する必要があります。ページングを避けるため、使用メモリの合計は、データベースキャッシュに割り当てられた分も含めて物理メモリのサイズを超えないようにしてください。どのような場合でも、`nsslapd-dbcachesize + nsslapd-cachememsize + slapd` プロセス自体で使用される固定 memory の累積値を、プロセス用のアドレス空間 4G バイトより大きくすることはできません。Nsslapd は 32 ビットアプリケーションです。

Sun Java System Directory Server は、リソースプール (接続およびスレッド) のサイズ指定に関して、検索動作の場合で多重度約 15 の高いパフォーマンスを示します。

perftune スクリプトにより、
 /var/opt/mps/serverroot/slapd-hostname/config/dse.ldif ファイルの ns-slapd スレッド化、db cache およびデータベースファイルシステムのマッピングを、次のようにチューニングします。

- dn:cn=config LDAP エントリの下位
 - nThreads に nsslapd-threadnumber 行を追加します。かなりの量のプロファイル変更 (LDAP 書き込み) が予想されない限り、ほとんどの場合デフォルト値 (30) で問題ありません。この場合、スクリプトは次の式を適用します。
 1 CPU の場合は nThreads = 30、2 CPU の場合は nThreads = 45、3 CPU の場合は nThreads = 60、4 CPU の場合は nThreads = 75。
 - nsslapd-accesslog-logging-enabled をオフに指定してアクセスログを無効化します。
- dn:cn=config,cn=ldbm database,cn=plugins,cn=config LDAP エントリの下位
 - /tmp/slapd-dsame1 に nsslapd-db-home-directory 行を追加します。
 - nsslapd-maxthreadsperconn 行を 20 に変更します。
 - nsslapd-dbcachesize 行を、newSize に変更します。ここで、newSize = 1.2 * は /var/opt/mps/serverroot/slapd-hostname/db/userRoot の下に配置されたすべての db3 ファイルのサイズです。
- dn:cn=userRoot,cn=ldbm database,cn=plugins,cn=config LDAP エントリの下位で、nsslapd-cachememsize 行を newSize に変更します。ここで、newSize = 3 * は id2entry.db3 のサイズです。
- NEED_REBOOT が yes の場合は、nsslapd-maxdescriptors を 16384 に設定します。

注 /var/opt/mps/severroot/slapd-hostname/config/dse.ldif ファイル
 には、次の形式でバックアップが作成されます。

filename-orig-date-pid

Sun Java System Web Server 6.1 のチューニング

次のセクションでは、`perftune` スクリプトによって JVM のチューニングを行い、Sun Java™ System Web Server for Portal Server のパフォーマンスを向上させる方法について説明します。

1. `WebServer-base/SUNWwbsvr/https-hostname/config` に配置された `magnus.conf` のパラメータを次のように指定します。
 - `RqThrottle 256`
 - `RqThrottleMin 128`
 - `StackSize 393216`
 - `ThreadIncrement 20`
 - `ConnQueueSize 4096`
 - `ListenQ 4096`

注 `WebServer-base/SUNWwbsvr/https-hostname/config/magnus.conf` ファイルには、次の形式でバックアップが作成されます。

`filename-orig-date-pid`

2. `WebServer-base//https-hostname//config` に配置された `server.xml` ファイルのパラメータを次のように指定して、JVM をチューニングします。
 - `-Xms3G` (この値はメモリが使用可能であれば 3G に設定します。それ以外の場合の `Xms` の値はデフォルトで 128 です。)
 - `-Xmx3G` (この値はメモリが使用可能であれば 3G に設定します。それ以外の場合の `Xmx` の値はデフォルトで 128 です。)
 - `-Xss128K`
 - `-Xloggc:/var/opt/SUNWappserver7/domains/domain1/server1/logs/gclog`
 - `-XX:NewSize=384M`
 - `-XX:MaxNewSize=384M`
 - `-XX:MaxPermSize=64M`
 - `-XX:PermSize=64M`
 - `-XX:+UseParNewGC`
 - `-XX:+UseConcMarkSweepGC`
 - `-XX:MaxTenuringThreshold=1`

- `-XX:SoftRefLRUPolicyMSPerMB=1`
- `-XX:+CMSClassUnloadingEnabled`
- `-XX:+CMSPermGenSweepingEnabled`
- `-XX:+PrintGCTimeStamps`
- `-XX:+ShowMessageBoxOnError`
- `-XX:+OverrideDefaultLibthread`
- `-XX:+DisableExplicitGC`
- `-XX:+PrintGCDetails`
- `-XX:+PrintClassHistogram`

Sun Java System Application Server 7.0 の チューニング

Sun™ Java System Application Server に Portal Server を配備する場合、アプリケーションサーバーインスタンスの最小および最大ヒープサイズは 3G バイトに設定されます。

`perftune` スクリプトには、JVM に 3G バイトのヒープを使用できないシステムのチューニングを避けるための安全手段が組み込まれています。

次のセクションでは、`perftune` スクリプトによって JVM のチューニングを行い、Sun Java™ System Application Server for Portal Server のパフォーマンスを向上させる方法について説明します。

1. `Deplaoy_Domain/Deploy_Instance/config` に配置された `init.conf` のパラメータを、次のように指定します。
 - `RqThrottle 256`
 - `RqThrottleMin 128`
 - `StackSize 393216`
 - `ThreadIncrement 20`
 - `ConnQueueSize 4096`
 - `ListenQ 4096`

注 `Deplaoy_Domain/Deploy_Instance/config/init.conf` ファイルには、次の形式でバックアップが作成されます。

`filename-orig-date-pid`

2. *Deplaoy_Domain/Deploy_Instance/config/sever.xml* にある JVM パラメータを、次のように指定します。
 - -Xms3G (この値はメモリが使用可能であれば 3G に設定する。それ以外の場合の Xms の値はデフォルトで 128)
 - -Xmx3G (この値はメモリが使用可能であれば 3G に設定する。それ以外の場合の Xmx の値はデフォルトで 128)
 - -Xss128K
 - -Xloggc:/var/opt/SUNWappserver7/domains/domain1/server1/logs/gclog
 - -XX:NewSize=384M
 - -XX:MaxNewSize=384M
 - -XX:MaxPermSize=64M
 - -XX:PermSize=64M
 - -XX:+UseParNewGC
 - -XX:+UseConcMarkSweepGC
 - -XX:MaxTenuringThreshold=1
 - -XX:SoftRefLRUPolicyMSPerMB=1
 - -XX:+CMSClassUnloadingEnabled
 - -XX:+CMSPermGenSweepingEnabled
 - -XX:+PrintGCTimeStamps
 - -XX:+ShowMessageBoxOnError
 - -XX:+OverrideDefaultLibthread
 - -XX:+DisableExplicitGC
 - -XX:+PrintGCDetails
 - -XX:+PrintClassHistogram

注 *Deplaoy_Domain/Deploy_Instance/config/server.xml* ファイルには、次の形式でバックアップが作成されます。

filename-orig-date-pid

注 `Deplaoy_Domain/Deploy_Instance/config/server.policy` ファイルに `-Djava.security.policy` が存在する場合は、`-Djava.security.policy` が `-Djava.security.policy=Deplaoy_Domain/Deploy_Instance/config/server.policy.NEVERUSED` によって置き換えられます。

その他の Sun Java System Application Server パラメータの設定によるゲートウェイの信頼性向上

Secure Remote Access を使用してパフォーマンスを最適化するには、次のように実装を設定します。

1. `AccessManager-base/SUNWam/lib/AmConfig.properties` ファイルに変更を加えて、アプリケーションサーバーの通知スレッドプールサイズを設定します。ファイルの先頭に次のような行があります。

```
Sun, Sun Microsystems, the Sun logo, and iPlanet
* are trademarks or registered trademarks of Sun Microsystems,
* Inc. in the United States and other countries.
```

次の行を追加して、スレッドプールのサイズを 200 に設定します。

```
/*Notification Thread Pool Size*/
com.ipplanet.am.notification.threadpool.size=200
```

2. ユーザー名 `amadmin` とインストール時に入力したパスワードを使用して、Portal Server 管理コンソールにログインします。
3. 「表示」メニューから「サービス管理」を選択します。
4. 「SRA 設定」、「ゲートウェイ」の順に選択します。
5. Default サーバーを選択し、「編集」をクリックします。
6. 「HTTP 接続を有効」チェックボックスにチェックマークを付けます。
7. 「HTTP ポート」フィールドに 80 と入力し、「保存」をクリックします。
8. ブラウザの Web アドレスフィールドに `http://fullservername:port` と入力し、管理者 (`admin`) として Sun Java System Application Server 管理コンソールにログインします。デフォルトポートは 4848 です。インストール時に入力したパスワードを使用します。
9. Access Manager をインストールしたアプリケーションサーバーインスタンスを選択します。
10. 「JVM 設定」をクリックし、「JVM オプション」をクリックします。
11. 「JVM オプション」フィールドに次の文字列を入力します。

```
-Dhttp.keepAlive=false
```

12. 「追加」をクリックし、「保存」をクリックします。
13. Portal Server をインストールしたアプリケーションサーバーインスタンスを選択します。
右のパネルに設定が変更されたことが表示されます。
14. 「変更を適用」をクリックします。
15. 「再起動」をクリックします。
16. アプリケーションサーバーが自動的に再起動します。
17. ゲートウェイがインストールされているサーバーで /opt/SUNWps/bin/perf ディレクトリに移動し、次の文字列を入力して、Secure Remote Access のチューニングパラメータを設定するスクリプトを実行します。

```
./perftune
```

18. AccessManager-base/SUNWam/lib/AmConfig.properties ファイルに変更を加えて、ゲートウェイの通知スレッドプールサイズを設定します。ファイルの先頭に次のような行があります。

```
Sun, Sun Microsystems, the Sun logo, and iPlanet
* are trademarks or registered trademarks of Sun Microsystems,
* Inc. in the United States and other countries.
```

次の行を追加して、スレッドプールのサイズを 200 に設定します。

```
/*Notification Thread Pool Size*/
com.iplanet.am.notification.threadpool.size=200
```

19. /opt/SUNWps/bin ディレクトリに移動します。ゲートウェイファイルに変更を加えて -Dhttp.keepAlive オプションを false に設定し、-Xms と -Xmx のヒープサイズオプションの設定を増大させます。
20. CMD 設定のオプションを次のように定義します。

注 CMD 設定は 1 行で定義してください。

```
CMD="$JAVA_HOME/bin/java -server -Xms3G -Xmx3G
-XX:+OverrideDefaultLibthread -Xss128K
-XX:MaxPermSize=64M -XX:PermSize=64M -XX:MaxNewSize=512M
-XX:NewSize=512M -XX:+UseParNewGC -XX:+UseConcMarkSweepGC
-XX:MaxTenuringThreshold=1
-XX:SoftRefLRUPolicyMSPerMB=1
```

```
-XX:+CMSClassUnloadingEnabled -XX:+CMSPermGenSweepingEnabled
-XX:+PrintGCDetails
-XX:+PrintGCTimeStamps -XX:+PrintClassHistogram
-XX:+ShowMessageBoxOnError -XX:+DisableExplicitGC
-Xloggc:/var/opt/SUNWps/debug/gclog.$GW_INSTANCE -classpath $CLASSPATH
$DEFINES1 $DEFINES2 $DEFINES3 $DEFINES4 $DEFINES5
$PROXY_DEFINES $BOOT_CLASSPATH com.sun.portal.netlet.EProxy"
```

21. /etc/opt/SUNWps/platform.conf.default ファイルに変更を加え、次のように `gateway.protocol` パラメータを `http` に、`gateway.port` パラメータをポート `80` に設定します。

```
gateway.protocol=http
gateway.port=80
```

22. 次のコマンドを入力してゲートウェイを再起動し、変更を有効にします。

```
PortalServer-base/SUNWps/bin/gateway -n default start
```

ここで、`default` はインストール時に作成されたデフォルトゲートウェイプロファイルです。

Portal Server デスクトップのチューニング

`caller` パラメータは、プロバイダを通してコンテンツを提供するためのスレッドプールのサイズを指定するために使用します。`caller` プールは、サイズ `0` に初期化されません。使用されて返されるときに、プールに項目が追加されます。`caller` プールは非常に大きなサイズに拡張可能ですが、通常はユーザーのポータルデスクトップのチャンネル数程度の大きさです。`sid` が同じ複数の並行スレッドが存在する場合、プールは $n * m$ のサイズに拡大可能です。ここで、`n` は同じ `sid` を持つ並行スレッドの数、`m` は指定された `sid` に対応するポータルデスクトップのチャンネル数です。

`perftune` スクリプトは、/etc/opt/SUNWps/desktop/desktopconfig.properties ファイルにある次のパラメータを変更し、プロバイダ `Caller` リソースプールを最適化します。

- `callerPoolMinSize` を `128` に増やします。
- `callerPoolMaxSize` を `512` に増やします。
- `callerPoolPartitionSize` を `16` に増やします。
- `templateScanInterval` を `3600` に増やします。

長時間のテストを実行する場合にポータルデスクトップの `caller` スレッドが生成され、これによって増える不要なメモリ使用量を最小限に抑えるには、これらのプロパティ (`templateScanInterval` を除く) を最初のデフォルト値に戻してください。

各プロパティを次のように変更します。

- callerPoolMinSize を 0 に変更します。
- callerPoolMaxSize を 0 に変更します。
- callerPoolPartitionSize を 0 に変更します。
- templateScanInterval プロパティを 30 から 3600 に増やします。

SSO アダプタテンプレートと設定

この付録では、SSO (シングルサインオン) アダプタを設定して、エンドユーザーが利用できるオプションを調節する方法について説明します。

この付録で説明する内容は次のとおりです。

- [シングルサインオンアダプタの概要](#)
- [SSO アダプタ属性ページ](#)

シングルサインオンアダプタの概要

シングルサインオンアダプタサービスを使用することで、エンドユーザーは Portal Server プロバイダやその他の Web アプリケーションなどのアプリケーションを使用し、1回のサインオンだけで各種リソースサーバーに認証アクセスすることができます。アクセスできるリソースサーバーは、システムに実装されている SSO アダプタインタフェースによって異なります。現時点では、Sun™ Java System Portal Server はアドレス帳、カレンダー、メールのリソースサーバーに対応した SSO アダプタを提供しています。Instant Messaging チャネルのシングルサインオンは、SSO アダプタ経由では利用できませんが、Sun Java System Identity Server 認証方法経由で利用できます。この方法については、[382 ページの表 17-1](#) の「authMethod プロパティ」を参照してください。アドレス帳、カレンダー、メールの各サービスは、次の製品で利用できます。

- Sun™ Java System Calendar Server 5.1.1、6.0、6 2004Q2
- Sun™ Java System Messaging Server 5.2、6.0、6 2004Q2

リソースサーバーには、通常はメールサーバー用の JavaMail など、標準的な API (アプリケーションプログラミングインタフェース) を使用して、アプリケーションによってアクセスされます。API を使用して認証接続を作成するには、API が接続の設定データを提供する必要があります。SSO アダプタの目的は、この設定データを供給することであり、データの格納には SSO アダプタサービスが使用されます。

SSO アダプタサービスは、テンプレートと設定という 2 つのレベルのデータを定義します。SSO アダプタテンプレートは、ユーザーが利用することになる接続のクラスを定義します。複数のユーザーが 1 つのテンプレートを使用できます。テンプレートは、そのテンプレートを使用するすべてのユーザーの同一のデータ値を定義します。これには、デフォルト値と、ユーザーが編集できる値に関する情報が含まれます。このため、SSO アダプタテンプレートは、グローバルサービスレベルで定義されます。

SSO アダプタ設定は、テンプレートの設定に加えて、組織、ロール、またはユーザーに固有のデータ値を提供します。設定はテンプレートを参照し、ユーザーが編集できないプロパティのデータ値をテンプレートから取得します。エンドユーザーが、SSO アダプタ設定の編集可能なプロパティを変更すると、その設定は、そのエンドユーザーだけに適用されます。

SSO アダプタサービスを使用する Sun Java System Portal Server の通信チャンネルは、テンプレートまたは設定のいずれかを参照し、リソースサーバーとの接続に必要なデータ値を取得します。チャンネルがテンプレートを参照する場合、ユーザーが設定情報を保存と、チャンネルは設定を参照するようになります。参照された設定は、テンプレートを参照します。

SSO アダプタ属性ページ

「SSO 属性」ページを使用して、次の管理を実行できます。

- SSO アダプタテンプレート : SSO アダプタテンプレートの作成、削除、またはプロパティの変更が可能。
- SSO アダプタの設定 : SSO アダプタの設定の作成、削除、またはプロパティの変更が可能。

[SSO アダプタテンプレートを作成する](#)

[SSO アダプタの設定を作成する](#)

[SSO アダプタテンプレートのプロパティを編集する](#)

[SSO アダプタの設定プロパティを編集する](#)

SSO アダプタテンプレートを作成する

1. Access Manager 管理コンソールにログインします。
2. 「サービス設定」タブを選択します。
3. 「SSO アダプタ」から「新規」をクリックして、SSO アダプタテンプレートを追加します。
「新規テンプレート」ページが表示されます。
4. テンプレート名を入力し、メニューから既存のテンプレートを選択します。
5. 「作成」をクリックします。
「テンプレートプロパティ」ページが表示されます。
6. 必要に応じてプロパティを修正します。
7. 完了したら、「保存」を選択します。

SSO アダプタの設定を作成する

1. 「SSO アダプタの設定」から「新規」をクリックして、SSO アダプタの設定を追加します。
「新規設定」ページが表示されます。
2. 設定名を入力し、メニューから SSO アダプタテンプレートを選択します。
3. 「作成」をクリックします。
「設定プロパティ」ページが表示されます。
4. 必要に応じてプロパティを修正します。

注 デフォルトのホスト名 MAIL_HOST (DNS 名または IP-Address) を入力するか、空白のままにします。

5. 完了したら、「保存」を選択します。

SSO アダプタテンプレートのプロパティを編集する

SSO アダプタテンプレートには、次のプロパティタイプがあります。

- 新規デフォルト: ユーザーのためにシステムによって提供されるプロパティを作成、選択、または削除可能。
 - 新規マージ: ユーザーによって提供されるプロパティを作成、選択、または削除可能。
 - 新規エンコード: エンコードされたプロパティを選択または削除可能
1. 変更する SSO アダプタテンプレートの隣の「プロパティの編集」リンクを選択します。
「テンプレートプロパティ」ページが表示されます。
 2. 必要に応じてプロパティを修正します。
「タイプ」の下にあるリンクをクリックして、プロパティタイプを変更できます。
 3. 完了したら、「保存」を選択します。

SSO アダプタの設定プロパティを編集する

1. 変更する SSO アダプタの設定の隣の「プロパティの編集」リンクを選択します。
「設定プロパティ」ページが表示されます。
2. 必要に応じてプロパティを修正します。
3. 完了したら、「保存」を選択します。

用語集

このマニュアルセットで使用される用語の完全なリストについては、『Java Enterprise System 用語集』(<http://docs.sun.com/source/819-1933?l=ja>) を参照してください。

索引

A

ACI, 103, 104
 委任管理者ロール, 145
 設定の定義, 147
amadmin, 95
amconsole, 98
amserver, 98
assign-source
 ロボットアプリケーション関数, 343
assign-type-by-extension
 ロボットアプリケーション関数, 343
authMethod プロパティ, 381, 453
authUsernameAttr, 381, 382

C

clear-source
 ロボットアプリケーション関数, 344
clientport, 420, 421
clientProtocol, 420, 421
clientRunMode, 381, 382
codebase, 381, 382
configDesc 属性, 402
contactGroup, 381, 382
cscal, 400
csuser, 400

D

db2ldif, 430
DIT, 103
DN, 205, 213, 218
dpadmin, 95, 241
 dryrun, 232, 242
 file 引数, 241
 list, 233, 242, 245
 modify, 242, 245, 251
 modify、combine, 248, 252, 253
 name オプション, 243, 245
 parent オプション, 243
 remove, 233, 242, 250, 251, 252, 253
 ガイドライン, 243
 グローバル, 242
 追加, 242, 247
dp-anon.xml, 229
dpEditAttributes
 「ディスプレイプロファイルのコレクション」を
 参照
dp-org-final.xml, 229
dp-org.xml, 229, 231
dp-providers.xml, 229
dryrun, 232, 242
DTD
 属性, 205
DTD の advanced 属性, 205

E

enablePerRequestConnection, 421
enableProxyAuth, 398, 421
encoded プロパティタイプ, 421

F

filter-by-exact
 ロボットアプリケーション関数, 339
filter-by-max
 ロボットアプリケーション関数, 340
filterrules-setup
 ロボットアプリケーション関数, 337
fuse, 219, 223

H

HTML
 リライタ JavaScript トークンルール, 273
 リライタアプレットルール, 274
 リライタ属性ルール, 272
 リライタフォームルール, 274
 リライタルール, 272
HTML テンプレート, 391 ~ 393
HTTPS プロトコル, 380, 419 ~ 424
HTTP プロトコル, 397, 419

I

IBM Lotus Notes, 374
IBM Lotus Notes サーバー, 375, 396, 404, 411 ~ 418, 419
idsvr, 381, 382
IMAP サーバーポート, 388
IMAP プロトコル, 397
IMProvider, 380
IM を有効にする, 380

Instant Messaging チャネル, 374, 380 ~ 387
 連絡先一覧, 390
iPlanet Directory Server Access Management Edition
 管理, 42
ipsadmin, 95

J

Java Plugin, 390
JavaScript
 DHTML パラメータ, 278
 DHTML 変数, 276
 DJS パラメータ, 278
 DJS 変数, 276
 EXPRESSION 変数, 276
 URL パラメータ, 277
 関数パラメータ, 277
 システム変数, 276
 リライタ変数ルール, 275
 リライタルール, 273, 275
Java Web Start, 382, 390
jnlp, 382
JSPProvider, 380
JSP 起動ページ, 383
JSP ファイル, 380

L

LDAP, 196, 205, 213, 218, 226
 認証, 124
 認証の設定, 128
ldapmodify
 ACI の定義, 147
LDAP プロトコル, 397
ldif2db, 431
Line of Business, 144
Linux 用のデフォルトのベースディレクトリ, 27
Lotus Notes Server
 「IBM Lotus Notes サーバー」を参照

M

MAIL-TYPE, 421

merge, 205

merge プロパティタイプ, 421

Microsoft Exchange Server, 374, 375, 396, 404 ~ 406, 419

Microsoft Outlook Web Access ソリューション, 404

modify

チャンネル, 239

multiplexor, 382, 390

mux, 381, 382

N

NCSO.jar ファイル, 413 ~ 418

netletRule, 381, 382

Netlet ルール, 385 ~ 386

NetMail, 102, 257

概要, 257

説明, 39

属性の変更, 262

リモートアドレス帳の使用, 264

NetMail Lite, 257

新規ウィンドウの表示設定, 263

O

ocxhost.zip ファイル, 405

Outlook

「Microsoft Outlook Web Access ソリューション」
を参照

P

par

ファイルのインポート, 178

ファイルのエクスポート, 178

ファイルの管理, 178

ファイルの作成, 178

ファイルの修正, 179

ファイルの配備, 178

par ファイル, 178

password, 421

perftune, 439, 441

plugin, 381, 382

POP プロトコル, 397

Portal Server

インストーラ, 375

パッケージ, 375

proxyAdminPassword, 396, 398, 399, 421

proxyAdminUid, 398, 399, 421

proxyAdminUid 属性, 396

R

RD, 286, 304

期限切れ, 310

データベースのインデックス再作成, 309

データベースの破棄, 310

データベース分析の表示, 308

RD エディタ, 305

remove

チャンネル, 225, 240, 251

プロバイダ, 250

プロパティ, 250

マージタイプ, 219, 220, 225

rwadmin, 95

S

Secure Remote Access

「SRA」を参照

service.http.allowadminproxy, 399

setup-regex-cache

ロボットアプリケーション関数, 337

setup-type-by-extension

- ロボットアプリケーション関数, 338
- smtpServer, 421
- SMTP サーバー名, 389
- Solaris
 - サポート, 30
 - チューニング, 441
 - パッチ, 30
- SRA, 382
- SSL
 - Directory Server の設定, 425
 - Portal Server のインスタンスの設定, 428
 - Portal Server の設定, 425, 426
 - サポートする Portal Server の変更, 427
 - 収集のためのリライトの設定, 280
- SSL (Secure Sockets Layer), 425
- SSO, 382, 404
- ssoClassName, 421
- ssoEditAttributes
 - 「ディスプレイプロファイルのコレクション」を参照
- SSO アダプタサービス, 392, 454
- SSO アダプタ設定, 395, 396, 401, 421 ~ ??, 454 ~ ??
- SSO アダプタテンプレート, 387 ~ ??, 391, 395, 396, 397 ~ ??, 419 ~ ??, 454 ~ ??
- subType, 421
 - sun-one, 421
- Summary Object Interchange Format (SOIF), 286
- sun-one, 421
- Sun ONE Directory Server
 - チューニング, 444
- Sun ONE Identity Server
 - 管理, 90
 - サービス, 102
 - 制約, 96
 - ツリー, 102
- Sun ONE Portal Server
 - インスタンスの削除, 72
 - 管理コンソールへのアクセス, 68
 - チューニング方法, 439
 - デスクトップのチューニング, 451
 - 複数インスタンスの作成, 71

- Sun ONE Web Server
 - チューニング, 446, 447
- SUNWiimps パッケージ, 375
- SUNWpsap パッケージ, 375
- SUNWpscp パッケージ, 375
- SUNWpsmp パッケージ, 375
- SUNWpssso パッケージ, 375

T

- type, 421

U

- uid, 398, 421
- UNIX
 - 認証の設定, 134, 135
- URL, 377, 380
 - ゲートウェイアドレスのプレフィックス指定, 269
 - 検索サーバーの定義, 290
 - 収集, 268
 - プレフィックス, 382
 - ポータル, 98
 - リダイレクトログイン, 169
- URLScrapperProvider, 268
 - 制限, 268
- URL の収集, 268
- userAttribute, 396, 398, 399

W

- Web コンテナ, 376, 403, 414 ~ 419

X

- XML, 392, 394 ~ 395
 - タグテキスト, 278
 - リライター属性, 279
 - リライタルール, 278
- XML ヘッダー、適切な, 241

あ

- アクセス権, 103
- アクセス制御命令 (ACI), 103, 104
 - 委任管理者ロール, 145
- アップロード
 - ディスプレイプロファイル, 232
 - リライタルールセット, 283
- アドレス帳チャンネル, 374, 387 ~ ??, 396, 397, 404
- アプリケーション設定の編集, 391 ~ 395, 396
- アプリケーションチャンネル, 383, 384
- アプレット
 - リライタルール, 274

い

- 委任された管理者, 226
- インスタンス
 - SSL の設定, 428
 - プロキシを使用する設定, 433
- インスタントメッセージの起動方法
 - Java Plugin, 390
 - Java Web Start, 390
- インデックスの再作成
 - データベース, 309
- インポート
 - エージェントの作成, 302
 - エージェントの編集, 304
 - 検索データベース, 302

え

- エンドユーザー
 - 信用情報, 396, 404

か

- カーネルのチューニング, 441
- 確認、デスクトップログファイル, 172
- 格納方式, 160
- カテゴリ
 - 検索, 287
 - 更新, 313
 - 子の作成, 312
 - 削除, 314
 - 設定, 312
 - 分類ルールの定義, 315
- カレンダーチャンネル, 374, 379, 397, 400 ~ 403, 404
- 監視
 - 検索状況, 292
- 管理
 - par ファイル, 178
 - 委任, 143
 - 委任されたロールの制限の設定, 154
 - 委任の設定, 147
 - 委任のためのモデルの開発, 146
 - 委任のためのロール, 144
 - 委任ロールの作成, 152
 - 委任ロールの割り当て, 153
 - インタフェース, 95
 - カテゴリ, 312
 - 検索, 285, 291
 - 検索操作, 291
 - 検索データベース, 302
 - 検索ロボット, 293
 - コンソール, 42
 - データベースの分類, 312
 - デスクトップ, 157, 164, 181
 - ユーザー, 102
 - リライター, 280
- 管理コンソール

- ナビゲーション, 42
- ログイン, 98
- 管理者の信用情報, 396
- 管理者のプロキシ認証, 373, 379, 396 ~ 399, 404
- 管理の委任, 143
 - 設定, 147
 - モデル, 146
 - 用語, 143
 - ロール, 144
 - ロールの作成, 152
 - ロールの制限の設定, 154
 - ロールの割り当て, 153

き

- キーワード
 - ユーザー, 213
- 期限切れ
 - データベース, 310
- 起動
 - Instant Messenger, 374
 - Mail, 374, 404
 - Portal Server, 100
 - アドレス帳, 374, 404
 - カレンダー, 374, 404
- 起動方法
 - Java Plugin, 390
 - Java Web Start, 390
- 起動ボタン, 374, 379

く

- クライアントポート, 389
- グローバル
 - 属性, 93, 170, 171
 - ディスプレイプロファイル, 227
 - レベル, 242

け

- 計画
 - 組織, 103
- ゲートウェイ
 - リライタによる変換, 269
- 検索, 102
 - インポートエージェントの作成, 302
 - インポートエージェントの編集, 304
 - 概要, 285
 - カテゴリ, 287
 - 管理, 285, 291
 - 基本設定, 291
 - サーバー URL の定義, 290
 - 状況の監視, 292
 - 詳細設定, 292
 - 設定, 288
 - 設定の表示, 291
 - 説明, 38
 - データベース, 286
 - データベースのインポート, 302
 - データベースの管理, 302
 - 分類, 287
 - ロボット, 286
 - ロボットの管理, 293

こ

- 更新
 - カテゴリ, 313
 - ディスプレイプロファイル, 243
- コレクション
 - 追加, 249
- コンテナ, 158, 196, 228, 234, 240
 - 階層構造, 159
 - チャンネル, 158
- コンテンツプロバイダ, 161

ナ

- サーバーポート, 389, 390
- サーバー名, 388, 389, 390
- サーバーリスト
 - Portal Server の追加, 433
- サービス
 - NetMail, 102
 - Sun ONE Identity Server, 102
 - 管理, 89, 91
 - 検索, 102
 - サービステンプレートの作成, 110
 - デスクトップ, 102
 - ライター, 102
- 最上位組織, 102
- サイトプローブ, 299
 - ロボットの実行, 300
- 削除
 - カテゴリ, 314
- 作成
 - par ファイル, 178
 - 委任管理ロール, 152
 - 検索データベースのインポートエージェント, 302
 - 子カテゴリ, 312
 - サービステンプレート, 110
 - サブ組織, 108, 109
 - 組織, 108, 109
 - ロール, 113, 149
- サブ組織, 103
 - 作成, 108, 109
 - 定義のガイドライン, 104
- サポート
 - Solaris, 30
- サポート関数のフィルタリング
 - ロボットアプリケーション関数, 342
- サンプルディスプレイプロファイル
 - dp-anon.xml, 229
 - dp-org-final.xml, 229
 - dp-org.xml, 229, 231
 - dp-providers.xml, 229
- サンプルポータル, 230

シ

- 識別名, 205, 213, 218
- シミュレータ, 299
 - ロボットの実行, 300
- シャットダウン関数
 - ロボットアプリケーション関数, 355
- 手動によるロード
 - ディスプレイプロファイル, 230
- 巡回
 - ロボットの制御, 295
- 新規ユーザー, 388
- シングルサインオン
 - 「SSO」を参照
- シングルサインオン (SSO), 91
- シングルサインオンアダプタ, 453
- 信用情報, 396, 404

ス

- スキーマ
 - データベースエイリアスの定義, 307
 - データベースの編集, 305

セ

- 制御
 - ロボットの巡回, 295
- 生成関数
 - ロボットアプリケーション関数, 351
- 設定
 - Directory Server の SSL, 425
 - LDAP 認証, 128
 - NetMail, 42
 - Portal Server の SSL, 425, 426
 - Portal Server のインスタンスの SSL, 428
 - UNIX 認証, 134, 135
 - 委任管理ロールの制限, 154
 - カテゴリ, 312
 - 管理の委任, 147

- 検索, 40
- 検索サービス, 288
- データベースの分類, 312
- データベースへのログイン, 435
- デスクトップ, 40
- 匿名認証, 129
- 認証, 124
- 認証メニュー, 127
- ファイルへのログイン, 434
- プロキシを使用するインスタンス, 433
- リライタ, 41

設定の説明フィールド, 398, 401, 422

セットアップ関数

- ロボットアプリケーション関数, 337

そ

送信済みフォルダへのコピー, 389

属性

- グローバル, 93, 170, 171
- 組織, 93
- ダイナミック, 93
- デスクトップの修正, 170, 171
- ポリシー, 93
- ユーザー, 93
- リライタ XML, 279
- ロボットのインデックス作成の定義, 298

組織, 102, 103

- 計画, 103
- 最上位, 102
- 作成, 108, 109
- 属性, 93
- 定義, 92

た

ダイナミック属性, 93

ダウンロード

- ディスプレイプロファイル, 232

- リライタルールセット, 282

タグテキスト

- リライタ, 278

タブ、新規, 247

ち

置換, 219, 222

- チャンネル, 245
- プロパティ, 246

チャンネル, 158, 195, 228, 234

- modify, 239
- remove, 225, 240, 251
- 置換, 245
- 追加, 238, 247
- 配備, 163
- パッケージ, 178
- 例, 162

チャンネル設定の例, 379

「チャンネルの編集」リンク, 234

チューニング

- Sun ONE Directory Server, 444
- Sun ONE Portal Server, 439
- Sun ONE Portal Server デスクトップ, 451
- Sun ONE Web Server, 446, 447
- TCP パラメータ, 442
- カーネル, 441

つ

追加

- Portal Server をサーバーリストに, 433
- コレクション, 249
- チャンネル, 238, 247
- プロパティ, 248

通信チャンネル, 379

- 設定の例, 379
- デフォルト設定, 379
- 複数インスタンス, 375, 376

編集ボタン, 373, 379, 388, 391
ツール
 ロボット, 299
ツリー
 階層構造, 104
 フラットな構造, 106

て

定義
 カテゴリ分類ルール, 315
 データベーススキーマのエイリアス, 307
 ロボットのインデックス作成の属性, 298
 ロボットのサイト, 294
停止
 Portal Server, 100
ディスプレイプロファイル, 164, 391 ~ 395
 default, 228
 root, 198, 220
 エラーメッセージ, 244
 階層構造, 213
 グローバル, 215, 227, 228, 230
 コンテナ, 196, 200, 228
 サブ組織, 228
 組織, 228
 ダイナミック, 228
 チャンネル, 195, 199, 228
 ディスプレイプロファイル, 229
 プロバイダ, 195, 199
 プロパティ, 196
 変更, 244
 編集, 254
 ディスプレイプロファイル, 218, 219
 ユーザー, 198
 優先順位, 213, 217, 218, 219, 224
 ロード, 228, 230
 ロール, 228
ディスプレイプロファイル属性, 394
 ソート, 392
 ソート順序, 392
ディスプレイプロファイルのコレクション
 dpEditAttributes, 391
 ssoEditAttributes, 391, 394, 396
ディスプレイプロファイルのマージ, 218, 219
 fuse, 219, 223
 remove, 219, 220
 置換, 219, 222
ディレクトリ情報ツリー (DIT), 91
データパネル, 44
データベース
 インデックスの再作成, 309
 期限切れ, 310
 検索, 286
 検索の RD の取得, 286
 検索のインポート, 302
 検索の管理, 302
 スキーマのエイリアスの定義, 307
 スキーマの編集, 305
 パーティション, 311
 分析の表示, 308
 分類, 287
 分類の管理, 312
 分類の設定, 312
 ロギング, 434
適切な XML ヘッダー, 241
デスクトップ, 102, 158
 概要, 157
 カスタマイズ, 162
 管理, 157, 164, 181
 グローバル属性, 171
 サービス属性の修正, 170, 171
 サービステンプレート, 226
 サブレット, 198
 説明, 38
 用語集, 157
 リダイレクトログイン, 169
 例, 35
 ログイン, 140, 172
 ログファイル, 172
デバッグ, 141
 Portal Server, 435
 ロボットツール, 299
デバッグレベル

- 設定, 436
- デフォルトのチャンネル設定, 379
- 伝達, 206
- テンプレート
 - 作成, 110

と

- 匿名認証
 - セッション方式, 130
 - 設定, 129
 - ユーザー ID 方式, 131
- ドメイン, 92

な

- ナビゲーションパネル, 44

に

- 認証
 - UNIX の設定, 134, 135
 - 管理, 89
 - コア, 125
 - 設定, 124
 - メニュー, 127
 - メンバーシップ, 125
- 認証なしの匿名デスクトップ
 - 「認証不要デスクトップ」を参照
- 認証不要デスクトップ, 400 ~ 403
- 認証方法, 453

は

- パーティション
 - RD データベース, 311

- 配備, 196
 - par ファイル, 178
 - チャンネル, 163
- 破棄
 - 期限切れ RD, 311
- バックアップ
 - Portal Server, 430
- パッケージ, 375
 - チャンネルとプロバイダ, 178

ひ

- 表示
 - 検索設定, 291
 - 製品情報, 99
 - データベース分析, 308
- ビルディングブロックプロバイダ, 161

ふ

- ファイル
 - エクスポート, 178
 - ディスプレイプロファイルのアップロード, 232
 - ディスプレイプロファイルのダウンロード, 232
 - ロギング, 434
- ファイルのエクスポート
 - par, 178
- フィルタ
 - ロボットの定義, 296
 - ロボットの定義の作成, 297
 - ロボットの定義の変更, 298
 - ロボットの定義の有効化, 298
 - ロボットのデフォルト, 296
- フィルタリング関数
 - ロボットアプリケーション関数, 339
- フォーム
 - リライタルール, 274
- 復元
 - Portal Server, 431

複数インスタンス, 375, 376

プロキシ認証

「管理者のプロキシ認証」を参照

プロバイダ, 195, 234, 237

remove, 250

アーカイブ, 163

パッケージ, 178

プロバイダアプリケーションプログラミングインタ
フェース (PAPI), 159

プロパティ, 196, 234

default, 202

remove, 250

入れ子, 207

階層構造, 203

グローバル, 202

コレクション, 204

参照, 204

整数, 204

置換, 246

追加, 248

伝達, 210

名前をつけない, 207

ブール型, 204

文字列, 204

プロパティタイプ

encoded, 421

merge, 421

分類, 287

へ

ベースドキュメント, 213

ヘッダー、適切な XML, 241

変更

NetMail の属性, 262

par ファイル, 179

SSL をサポートする Portal Server, 427

デスクトップサービス属性, 170, 171

ロボットフィルタ定義, 298

変更、優先順位, 251

編集, 304

RD, 304

検索データベースのインポートエージェント,
304

データベーススキーマ, 305

編集ボタン, 373, 379, 388, 391

ほ

ポータル

管理, 42

配備プラットフォーム, 34

ポータルデスクトップ, 388

通信チャネルの編集ボタン, 373, 379, 388, 391

ポリシー

管理, 91, 136

属性, 93

定義, 94

ま

マニュアル

概要, 28

む

無効化

ロボットの定義, 298

ロボットフィルタ定義, 298

め

命名属性, 396

メールチャネル, 374, 379, 380, 397, 404, 419

メンバーシップ

認証, 125

ゆ

有効化

ロボットフィルタ定義, 298

有効化パラメータ, 336

ユーザー

管理, 89, 90, 102

既存の有効化, 114

計画, 104

属性, 93

ユーザーパスワード, 389, 390

ユーザー名, 389, 390

優先順位, 206, 213, 217

同じ, 214

変更, 251

ユーティリティ

par, 178

ロボット, 299

よ

読み取り専用通信チャンネル, 400 ~ 403

り

リソース記述, 286

リダイレクト

ログイン, 169

リライト, 102, 267

DHTML パラメータ, 278

DJS パラメータ, 278

HTML 属性ルール, 272

HTML フォームルール, 274

JavaScript URL パラメータ, 277

JavaScript 関数パラメータ, 277

JavaScript ルール, 273, 275

SSL 用の URLscaper の設定, 280

XML コンテンツのルール, 278

XML 属性, 279

アプレットルール, 274

概要, 267

管理, 280

ゲートウェイ URL のプレフィックス指定, 269

サポートされる URL, 270

説明, 38

タグテキスト, 278

デフォルトルールセットの復元, 284

ルールセットのアップロード, 283

ルールセットの削除, 284

ルールセットの作成, 281

ルールセットのダウンロード, 283

ルールセットの編集, 282

ルールとルールセットの定義, 270

る

ルール

HTML リライター, 272

カテゴリ分類の定義, 315

リライト JavaScript, 275

リライト JavaScript トークン, 273

リライト XML コンテンツ, 278

リライトアプレット, 274

リライトの定義, 270

リライトフォーム, 274

ルールセット, 270

アップロード、リライト, 283

削除、リライト, 284

作成、リライト, 281

ダウンロード、リライト, 283

編集、リライト, 282

リライトデフォルトの復元, 284

れ

列挙関数

ロボットアプリケーション関数, 350

連絡先一覧, 390

ろ

- ロード
 - ディスプレイプロファイル, 228
- ロール, 104
 - 委任管理の作成, 152
 - 委任管理の割り当て, 153
 - 管理の委任, 144
 - 管理の委任に対する制限の設定, 154
 - 作成, 113, 149
 - 定義, 92
 - 定義のガイドライン, 104
 - 割り当て, 113, 150
- ロール管理者ロール, 144
- ロールツリー, 91
- ロギング, 141
 - 属性, 141
 - データベースへのログ記録の設定, 435
 - ファイルへのログ記録の設定, 434
- ロケーションパネル, 43
- ロック, 205, 217, 224
- ロボット, 286
 - インデックス作成属性の定義, 298
 - 管理, 293
 - サイトの定義, 294
 - サイトプロブユーティリティ, 300
 - シミュレーション, 299
 - シミュレータユーティリティ, 300
 - 巡回の制御, 295
 - フィルタ定義の作成, 297
 - フィルタ定義の変更, 298
 - フィルタ定義の無効化, 298
 - フィルタ定義の有効化, 298
 - フィルタの定義, 296
 - ユーティリティ, 299
- ロボットアプリケーション関数
 - サポート関数のフィルタリング, 342
 - シャットダウン関数, 355
 - 生成関数, 351
 - セットアップ関数, 337
 - フィルタリング関数, 339
 - 列挙関数, 350

わ

- 割り当て
 - 委任管理ロール, 153
 - ロール, 113, 150

